

- Aprenda la disciplina, ejerza el arte y contribuya con sus ideas en www.ArchitectureJournal.net
Recursos que sirven de base

Flujo de Trabajo Generacional

Crear Aplicaciones sobre una Plataforma de Flujo de Trabajo

La Metáfora de "The Amazing Race"

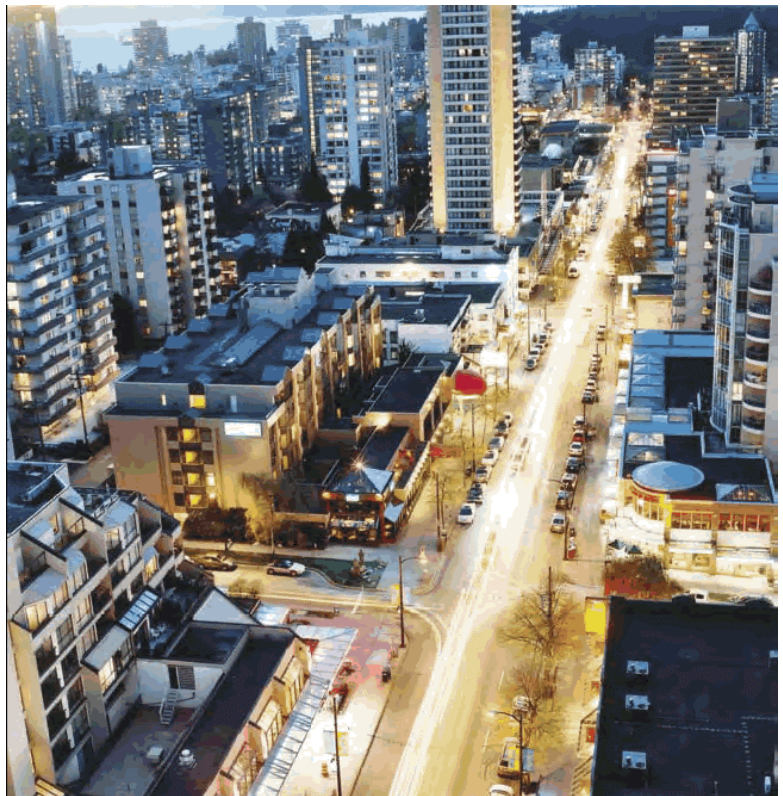
Explorar Arquitecturas para el Flujo de Trabajo Humano

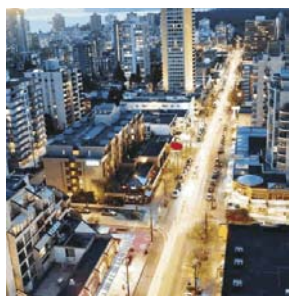
El Flujo de Trabajo en la Integración de Aplicaciones

Simplificar el Diseño de Flujos de Trabajo Complejos

Habilitación de la Empresa Orientada al Servicio

Modelación Orientada al Servicio para Sistemas Conectados





Contenidos

Prólogo

1

Por Simon Guest

Crear Aplicaciones sobre una Plataforma de Flujo de Trabajo

2

Por David Green

Un flujo de trabajo es importante para resolver problemas de negocios. Examine una variedad de aplicaciones que demuestran las decisiones y razonamientos que enfrentan los arquitectos cuando construyen una plataforma de flujo de trabajo.



La Metáfora de "The Amazing Race"

9

Por Vignesh Swaminathan

Las empresas de hoy en día se dan cuenta del potencial que tiene automatizar sus procesos de negocio. Averigüe sobre la gestión de procesos de negocio de alto nivel a través de la analogía de un juego de un show televisivo basado en la realidad.



Explorar Arquitecturas para el Flujo de Trabajo Humano

16

Por Jesus Rodríguez y Javier Mariscal

Existen dos componentes de sistemas de flujo de trabajo humano y patrones representativos de interacciones de procesos entre las personas y las actividades empresariales. Descubra la forma de aplicar estos componentes para implementar estos procesos.



El Flujo de Trabajo en la Integración de Aplicaciones

19

Por Kevin Francis

La integración de aplicaciones es uno de los mayores desafíos que enfrentan los arquitectos en la actualidad. Analice brevemente un sistema para la integración de una aplicación por medio del uso de herramientas tales como tecnologías de flujo de trabajo.

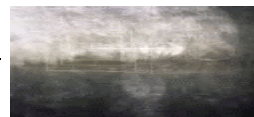


Simplificar el Diseño de Flujos de Trabajo Complejos

24

Por Andrew Neddleman

El diseño efectivo de flujos de trabajo requiere diversas técnicas que hacen que el flujo de trabajo sea un desafío aún para los arquitectos más expertos. Aprenda un método para simplificar el proceso de diseño de sistemas complejos utilizando un nuevo tipo de diagramación.



Habilitación de la Empresa Orientada al Servicio

27

Por William Oellermann

Si bien construir varios servicios de Web puede ser difícil, administrarlos puede ser realmente difícil. Explore el uso de un modelo que puede ayudarlo a planear capacidades para una empresa habilitada para la prestación de servicios.



Modelación Orientada al Servicio para Sistemas Conectados – Primera Parte

33

Por Arvindra Sehmi y Beat Schwegler

Los arquitectos desean identificar los artefactos de forma adecuada y en el nivel de abstracción correcto. Verifique en esta primera parte de una serie de dos artículos un enfoque para modelar sistemas conectados, orientados al servicio que fomenta una alineación estrecha entre las soluciones IT y las necesidades del negocio.



Fundador

Arvindra Sehmi
Microsoft Corporation

Jefe Editor

Simon Guest
Microsoft Corporation

Consejo Editorial de Microsoft

Gianpaolo Carraro
John deVadoss
Neil Hutson
Eugenio Pace
Javed Sikander
Philip Teale
Jon Tobey

Editor Comercial

Marty Collins
Microsoft Corporation

Diseño, Impresión y Distribución:

Publicaciones Técnicas Fawcette
Jeff Hadfield, VP de Publicación
Terrence O'Donnell, Editor Gerente
Michael Hollister, VP de Arte y producción
Karen Koenen, Director de Circulación
Brian Rogers, Director de Arte
Kathleen Sweeney Cygnarowicz,
Gerente de Producción



La información contenida en la revista *The Best of the Microsoft Architecture Journal* ("Journal") se brinda sólo con fines informativos. El material publicado en el *Journal* no constituye la opinión o asesoramiento de Microsoft y no debe basarse en ningún tipo de material publicado en ella sin antes buscar asesoramiento independiente. Microsoft no provee garantía o representación alguna respecto a la precisión o aptitud de los fines de cualquier material del *Journal* y en ningún caso Microsoft acepta responsabilidad de ningún tipo, incluyendo responsabilidad por culpa (excepto por daño contra los derechos personales del individuo o fallecimiento), por cualquier tipo de daños o perjuicios o pérdidas (incluyendo, sin limitación, pérdida del negocio, rédito, ganancias o pérdida consiguiente) de cualquier índole o naturaleza que resultare del uso del presente *Journal*. El *Journal* puede contener imprecisiones técnicas y errores de tipografía. El *Journal* se actualizará de vez en cuando y podrá otras veces estar desactualizado. Microsoft no acepta responsabilidad alguna por mantener la información de este *Journal* actualizada ni por el incumplimiento del hecho. Este *Journal* contiene material propuesto y creado por terceros. Hasta el alcance máximo permitido por la ley aplicable Microsoft excluye toda responsabilidad por cualquier acto ilegal que surgiera de un error, omisión o imprecisión en este *Journal* y Microsoft no se responsabiliza del material suministrado por terceros.

Todos los derechos del autor, marcas registradas y cualquier otro tipo de propiedad intelectual del material contenido en el *Journal* pertenecen y son licencia exclusiva de Microsoft Corporation. Queda totalmente prohibida la copia, reproducción, transmisión, almacenamiento, adaptación o modificación de la forma o contenido del presente *Journal* sin previo consentimiento por escrito por parte de Microsoft Corporation y los autores individuales.

© 2005 Microsoft Corporation. Todos los derechos reservados.

Prólogo

Estimado arquitecto:

Recuerdo bien esa mañana. El día estaba despejado y frío, y yo estaba en las oficinas de Microsoft, en Londres, sentado con Arvindra Sehmi mientras me explicaba su visión y me mostraba uno de los primeros magníficos prototipos del *Journal*. Como bien sabrá, desde esa mañana en adelante el *Journal* ha ido cada vez mejor. Usted está leyendo la 7ma edición del *Journal*, que llega a 30.000 abonados en todo el mundo –tanto en su formato impreso como en su formato online a través de nuestra nueva página, ArchitectureJournal.net que se lanzó el mes pasado. Mientras charlaba con Vin, nunca imaginé que tres años después yo iba a tener el honor de asumir la responsabilidad como editor de la revista.

Si bien no hay cambios importantes en el formato, como editor tengo planeado introducir algunas nuevas directivas para *The Architecture Journal*. Primero y en base a la creciente suscripción, traduciremos esta revista en ocho idiomas para todo el mundo. Si usted está leyendo esta edición en otro idioma que no es el inglés, me gustaría ser el primero en darle la bienvenida a este nuevo formato y espero que se establezca como una tendencia continua.

Segundo, a partir de esta edición, avanzaremos hacia un enfoque temático en el que la mayoría de los artículos de cada edición siguen un patrón o tema en común. Como ya habrá visto en la tapa, el tema de esta edición es "Flujo de Trabajo Generacional". Para comenzar con el tema, empezamos con un artículo de Dave Green, arquitecto de nuestro producto Windows Workflow Foundation (WF). Dave comparte las decisiones y razonamientos que enfrentó su equipo al construir una plataforma de flujo de trabajo, que en última instancia, dieron forma al producto que están creando.

Luego tenemos a Vignesh Swaminathan, gerente de producto en CORDES R&D. En su artículo, Vignesh lo llevará por una "Amazing Race" y analizará las similitudes entre las matrices de decisiones para el flujo de trabajo en una empresa y un show televisivo popular. Para agregarle un toque humano al *Journal*, Jesus Rodríguez y Javier Mariscal tratarán los componentes principales de los sistemas de flujo de trabajo humano y describirán los patrones que pueden utilizarse para modelar las tareas relacionadas con las personas.

Para conectar todas estas piezas, Kevin Francis analizará la integración de aplicaciones con un flujo de trabajo en su artículo "El Flujo de trabajo en la integración de aplicaciones". Y para completar el conjunto de artículos sobre flujo de trabajo de esta edición, Andrew Needleman nos mostrará la técnica que utiliza para simplificar el proceso de comunicación de los flujos de trabajo complejos con los expertos de negocios denominada "puntos y líneas".

Siguiendo los artículos sobre flujo de trabajo, tenemos la suerte de contar con el artículo de William Oellermann sobre el Maduración de Orientación al Servicio para Empresas (ESOMM), un modelo de maduración que analiza la gestión de servicios en la empresa. Finalmente, para cerrar esta edición, hemos decidido no perdonar a Arvindra completamente y él regresa junto con Beat Schwegler para ofrecernos la Primera Parte de un excelente análisis de la "Modelación orientada al servicio para sistemas conectados". La segunda parte se publicará en la 8va edición de este *Journal*.

En términos generales, ha sido una gran experiencia organizar mi primer ejemplar, y espero que encuentre los artículos útiles y que lo inviten a la reflexión. Como nuevo editor, una vez más me gustaría darle la bienvenida y espero pronto tener noticias de varios de ustedes.

Simon Guest



Crear Aplicaciones sobre una Plataforma de Flujo de Trabajo

Por David Green

Síntesis

Un flujo de trabajo puede ser muy útil para resolver problemas de negocios. Aquí investigaremos y nos familiarizaremos con la idea de construir aplicaciones sobre una plataforma de flujo de trabajo. Esta plataforma admite conceptos claves y proporciona la base para construir aplicaciones estructuradas utilizando estos conceptos, incluyendo los productos de flujo de trabajo, hasta ahora comprendidos. Estudiaremos una variedad de aplicaciones para explorar las características necesarias de esta plataforma lo que conduce a un análisis de los beneficios potenciales al construir aplicaciones sobre una plataforma de flujo de trabajo. También analizaremos *Windows Workflow Foundation* como un medio para poner en práctica estos beneficios.

La noción de flujo de trabajo existe desde hace mucho tiempo y ha sido conveniente constantemente como una forma de combatir problemas del negocio. Los problemas a los que se les ha aplicado un método de flujo de trabajo, por lo general, han mostrado tres características: el valor clave de negocios que proporciona es la *coordinación*, por ejemplo, al organizar contribuciones múltiples para la preparación de un presupuesto o para orientar la revisión de un documento. Cada instancia del proceso de negocios afectado posee una *duración extensa*, que por lo general se mide en días, semanas o meses, más que en minutos. El proceso de negocios posee participantes *humanos*, quienes generalmente contribuyen con la mayor parte del producto de trabajo.

Sin embargo, sólo una pequeña cantidad de los problemas de negocios que poseen estas características pueden resolverse aplicando un enfoque de flujo de trabajo. Con frecuencia, el proceso de negocios no se registra en absoluto como información que puedan leer las máquinas. Sino más bien que cada persona que participa en el proceso de negocios interactúa con los sistemas de negocios que no conocen las semánticas del proceso como un todo, como por ejemplo un sistema de información del cliente, y con otros participantes humanos por medio de canales de comunicación del contenido neutral tales como correos electrónicos. Cada participante humano utiliza un modelo mental de su participación en el proceso de negocios total para determinar su comportamiento.

Las ventajas de crear un modelo del proceso de negocios que pueda ser leído por una máquina, es decir, un flujo de trabajo, no son difíciles de prever. Tres de los beneficios que puede proporcionar un flujo de trabajo son: introspección, monitoreo y optimización. Un conjunto de flujos de trabajo relacionados puede utilizarse para lograr la *introspección* dentro del flujo de trabajo a través de una organización. Para *monitorear*, es muy útil saber qué individuos aportan trabajo a qué proceso de negocios cuando se tratan de comprender los costos y trabajos

realizados. Para *optimizar*, contar con un modelo del trabajo que se emprende y poder utilizar el modelo para interpretar el comportamiento, posibilita el razonamiento acerca de la forma de optimizar el proceso del negocio.

Modelo de flujo de trabajo

Dados estos beneficios convincentes, ¿Por qué los modelos de flujo de trabajo no se han utilizado de forma más general? La respuesta más probable sería que el costo de su uso ha sido muy alto. Estos costos incluyen costos del *producto*, es decir, el costo directo de la adquisición de un producto de flujo de trabajo; costos de *integración*, en los que los procesos modelados como flujos de trabajo necesitan ser integrados como parte de un sistema de negocios más amplio y costos de *estandarización*, en los que resulta difícil para una gran organización estandarizar en base a una única tecnología de flujo de trabajo. Las variaciones en los productos de flujo de trabajo también indican que la portabilidad del modelo y las técnicas representan problemas.

Veamos la posibilidad de tratar estos problemas de bloqueo construyendo aplicaciones sobre una plataforma de flujo de trabajo que sea de bajo costo, ubicua, uniforme y de fácil integración en las aplicaciones. Para ser claro, la idea no es reemplazar productos de flujo de trabajo. Más bien, la hipótesis es que es útil desviar soporte para algunos conceptos centrales del flujo de trabajo e implantarlos en una plataforma sobre la cual se puedan construir productos de flujo de trabajo y otras aplicaciones (Ver Figura 1).

Un flujo de trabajo es un *modelo*, lo que implica una descripción del comportamiento del negocio que puede ser leída por una máquina y no es código. Más adelante, analizaremos el significado y beneficios de este concepto en el contexto del valor de una plataforma de flujo de trabajo.

Un modelo de flujo de trabajo describe una organización de *unidades de trabajo*. Por ejemplo, supongamos que el proceso de revisión de un documento especifica que Joe escribe el documento y luego Fred lo analiza. Aquí las unidades de trabajo son: en primer lugar la escritura y luego el análisis del documento, y la organización implica que una tarea debe seguir a la otra. Este concepto no es una idea radical. El código que realiza sucesivas llamadas a dos subrutinas es un ejemplo válido del concepto. El interés reside más bien en las formas que toma la organización.

Para evaluar la hipótesis de la plataforma de flujo de trabajo, consideraremos una variedad de aplicaciones del mundo real y exploraremos las características que tiene que poseer una plataforma de flujo de trabajo si debe probar que es útil.

El proceso de *revisión de un documento* toma como parámetros de entrada un conjunto de pares [revisor, función] que describen qué personas participan en el flujo de trabajo y en qué funciones. Algunos valores posibles para las funciones son: requerido, opcional, aprobador final y propietario. El proceso de revisión, por lo tanto, continúa hasta que todos los revisores hayan realizado las funciones asignadas y notifiquen el resultado al propietario.

Aquí los ítems de trabajo son las revisiones de los documentos organizadas por el proceso de revisión. Existen tres características interesantes para destacar, a saber: puntos múltiples de interacción, actividad automatizada y actividad humana y la necesidad de realizar un cambio dinámico.

Contratos de flujo de trabajo

El flujo de trabajo posee múltiples puntos de interacción o contratos. En primer lugar, existe un contrato con el revisor. Este contrato comprende el pedido de revisión del documento al revisor, la aceptación del veredicto y cualquier comentario de revisión, así como también, informarle al revisor que no se necesitan más sus aportes (si se cancela la revisión o tal vez, si la cantidad necesaria de revisores ha realizado una votación afirmativa). El contrato también permite al revisor delegar una revisión. Entonces, existe un segundo contrato con el revisor final, que es una especialización del contrato del revisor. En tercer lugar, existe un contrato con el propietario de la revisión que le permite al propietario cancelar la revisión y ser informado de los resultados finales de la revisión. Por último, existe un contrato con el iniciador del proceso de revisión quien inicia la revisión y proporciona los parámetros necesarios.

Es algo común que los flujos de trabajo conecten partes múltiples a través de varios contratos (Ver Figura 2). El flujo de trabajo de la revisión del documento es esencialmente un coordinador, iniciado por medio de un contrato que coordina una variedad de participantes a través de uno o más contratos adicionales.

El flujo de trabajo de la *revisión del documento* orienta la actividad humana. Sin embargo, también puede orientar las actividades automatizadas, tales como versiones de almacenamiento del documento en un depósito en la medida que la revisión progresa. Desde el punto de vista del flujo de trabajo, no existe una diferencia esencial. Puede considerarse que el flujo de trabajo por lo general se comunica con los servicios por medio de contratos. Un caso especial de un servicio representa otro flujo de trabajo. Otro caso especial es un humano. De varias formas, un humano es el servicio asíncrono original: uno nunca sabe cuándo responderá o aún si lo hará.

Una característica de este tipo de flujo de trabajo es que los participantes solicitarán cambios en la medida que el flujo de trabajo se ejecute. Por ejemplo, un revisor puede delegar una tarea de revisión a un colega o compartir con un subordinado el trabajo implicado en una tarea de revisión.

Existen dos formas de tratar este requerimiento. Una es construir una interpretación de todos los cambios posibles en el flujo de trabajo. Entonces, la solicitud de delegación se convierte simplemente en otra función del contrato entre el flujo de trabajo y el revisor. Otra posibilidad es ver el cambio como algo separado del flujo de trabajo, donde el cambio se implementa como una función externa que cambia el modelo del flujo de trabajo. En este enfoque, el resultado de la delegación es un nuevo modelo de flujo de trabajo idéntico a aquél en el que originalmente la tarea de revisión se asignó a un delegado.

La solicitud de un paso de aprobación adicional agregaría una nueva tarea de aprobación al modelo de flujo de trabajo, que bien puede no contener pasos de aprobación en su forma

original. El flujo de trabajo ya no debería anticipar más todas las modificaciones posibles, en el mejor de los casos, tendría que ocuparse de restringir las áreas del modelo que están sujetas a cambios.

Ambos enfoques son útiles. Construir un conocimiento dentro de un flujo de trabajo es fácil de modelar y comprender. La generalización de las operaciones es más compleja de modelar, pero a su vez es más eficaz y ágil.

En un caso extremo pero interesante del último enfoque, el flujo de trabajo comienza la ejecución con muy poco o ningún contenido y el comportamiento requerido se agrega de forma dinámica a través de los participantes del flujo de trabajo. Aquí, las operaciones disponibles para modificar el flujo de trabajo se convierten en vocabulario que el usuario puede utilizar para construir el comportamiento deseado en la medida que el flujo de trabajo progresa.

Colaboración problema-resolución

Para analizar un ejemplo específico de una aplicación de *colaboración problema-resolución*, consideremos un déficit de inventario. Una cadena de montaje marca un dispositivo y la computadora indica que existen suficientes productos en el stock para el propósito. Sin embargo, cuando el encargado del depósito va a buscar el producto para enviarlo a la cadena de montaje, descubre que existe un déficit de 10 unidades.

Se requiere la colaboración entre los encargados del depósito, el ejecutivo de cuentas de clientes y el gerente de producción para solucionar el problema. Cada función en la colaboración puede tomar acciones características. El departamento de suministros podría ordenar más productos, tal vez utilizando un proveedor diferente o pagando a un proveedor existente más dinero para que éste realice una entrega más rápida. El ejecutivo de cuentas podría contactar al cliente y proponerle una entrega diferida o dividir la entrega en dos partes asumiendo todos los costos extras del envío. El

“DE VARIAS FORMAS, UN HUMANO ES EL SERVICIO ASÍNCRONO ORIGINAL: UNO NUNCA SABE CUÁNDO RESPONDERÁ O AÚN SI LO HARÁ”.

gerente de producción podría desviar a otro cliente los dispositivos montados para una determinada orden. El encargado del depósito podría verificar su stock físico para tratar de encontrar los productos que faltan. Cada acción determinada puede realizarse múltiples veces.

Una limitación obvia es que la colaboración no está completa hasta que alguna combinación de las acciones precedentes resuelve el déficit. Con frecuencia, también existen limitaciones empresariales. Por ejemplo, puede existir una norma que establezca que no está permitido el desvío de entregas de Clientes Oro. También, las acciones los afectará mutuamente. Por ejemplo, puede existir una política que establezca que el costo adicional total de las medidas correctivas no puede exceder el 5 por ciento del costo de fábrica original. Por lo tanto, realizar un pedido de entrega más rápida a un costo mayor puede impedir que un envío sea dividido.

Figura 1 Flujo de trabajo monolítico para apilar

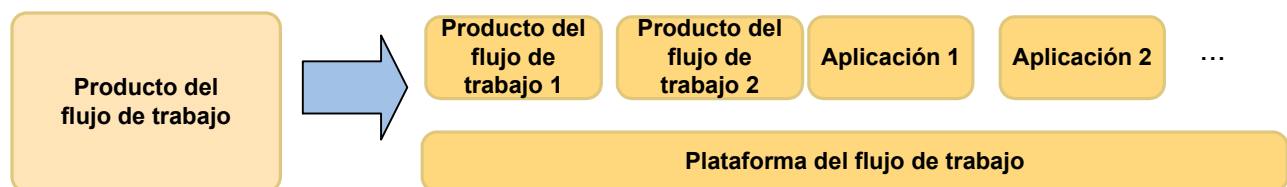
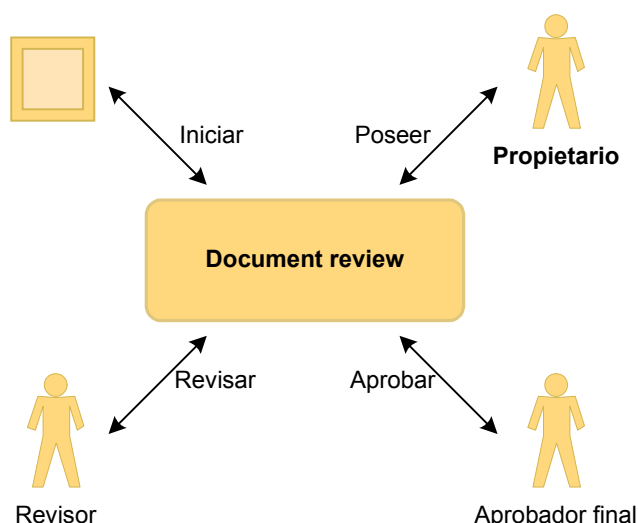


Figura 2: Diagrama de un contrato para la aplicación de revisión de un documento



En este caso los ítems de trabajo son las acciones que los varios participantes pueden tomar en la medida que buscan solucionar el déficit del inventario. La organización, sin embargo, no es la misma que la requerida en la revisión de un documento. Los participantes no son controlados; por el contrario, ellos eligen qué acciones realizar y cuándo realizarlas. No obstante, estas elecciones están limitadas por la organización del flujo de trabajo que posee dos aspectos: 1) Las acciones se focalizan en alcanzar una meta; en este caso, resolver el déficit del inventario. Cuando se inicia la resolución del problema se crea un espacio de colaboración limitado, y no se cierra hasta que se alcanza el objetivo. 2) Los participantes no son libres de realizar acciones arbitrarias. Más bien, las acciones disponibles están determinadas por la función que está cumpliendo el participante y por el estado de colaboración. El conjunto de acciones disponible lo determinan las políticas relacionadas a la meta y las políticas globales tales como la restricción sobre venta al descubierto de clientes preferenciales. Las acciones disponibles varían en la medida que la colaboración progresa.

La experiencia del participante no es más la de realizar las tareas asignadas. En cambio, un participante consulta las acciones que se encuentran disponibles para él en la actualidad, realiza una o ninguna de esas acciones y luego repite el ciclo.

Por lo tanto, aquí el principal requisito nuevo es para la forma de organización de los ítems de trabajo que está esencialmente orientada a la meta y al estado de los datos.

Operaciones con secuencias de comandos

Las operaciones con secuencias de comandos son simplemente un conjunto de operaciones que se componen utilizando una secuencia de comandos. Un ejemplo podría ser una herramienta de escritorio que permite a un usuario definir y ejecutar una serie de tareas comunes, como copiar archivos y comentarlos.

Sería poco común considerar el uso de un producto de flujo de trabajo típico para este propósito. Sin embargo, esto se ajusta al patrón de plataforma de flujo de trabajo de un conjunto de unidades de trabajo organizadas por un modelo. En este caso, el modelo es una secuencia, tal vez con soporte para la ejecución en bucles y condicional. Por lo tanto, si una plataforma de flujo de trabajo fuera lo suficientemente general y de bajo costo, sería posible considerar su aplicación a este tipo de problemas. ¿Agregaría esto algún valor?

Una característica de las operaciones con comandos de secuencias que las implementaciones típicas no contemplan en la actualidad es el problema del flujo de datos. Es común que los datos requeridos para una operación sean el resultado de alguna operación previa, pero esta información no está por lo general modelada en la secuencia de comandos. Por lo tanto, puede ser que un usuario que ensambla tareas utilizando una herramienta de escritorio no sea advertido cuando crea una secuencia de comandos que los datos de prerrequisito para una tarea no han sido provistos y sólo se dará cuenta del error cuando ejecute la secuencia de comandos. Un modelo de flujo de trabajo que pueda describir estas dependencias de datos podría agregar un valor más claro para los autores de secuencias de comandos.

Una estrategia sería simplemente incluir estructuras de flujo de datos en el modelo de flujo de trabajo. Es muy debatible que el modelo de flujo de trabajo básico necesite incluir características estructurales básicas como secuencias, condiciones y bucles, pero no está claro si el flujo de datos es lo suficientemente universal como para que lo representen elementos de primera clase del modelo.

Una estrategia alternativa es estratificar el soporte para el flujo de datos en función de un flujo de trabajo básico y extensible. Un modelo de flujo de trabajo que puede enriquecerse con abstracciones adecuadas para una variedad de dominios problemáticos se adecua muy bien a la noción de una plataforma de flujo de trabajo. Este enfoque evita tanto la complejidad que se crea al incluir en el modelo base una gran variedad de estructuras semánticas especializadas para problemas diferentes así como también las restricciones impuestas al limitar el modelo del flujo de trabajo a un conjunto fijo de estructuras.

Analicemos ahora una aplicación para un *usuario guiado*. Un ejemplo sería un sistema interactivo de respuesta de voz (IVR) y otro sería el sistema de un centro de atención telefónica que guía a los operadores por medio de secuencias de ventas o soporte.

“UN MODELO DE FLUJO DE TRABAJO QUE EXPRESA EL PROPÓSITO EMPRESARIAL CENTRAL DE LA APLICACIÓN, DESPROVISTO DE CUALQUIER MATERIAL TÉCNICO IRRELEVANTE, ES UNA FORMA EFECTIVA DE LOGRAR ESTA COMUNICACIÓN ENTRE EL PERSONAL EMPRESARIAL Y SU STAFF”.

La esencia de estas aplicaciones es guiar al usuario a través de una serie de operaciones necesarias para lograr su objetivo. La organización de estas operaciones por lo general es utilizada para orientar la presentación al usuario, ya sea una presentación generada o un conjunto de botones de comando activados o desactivados sobre un formulario.

Una característica de este tipo de aplicación es que el flujo de trabajo es la parte de la aplicación que cambia con más frecuencia. También, los patrocinadores empresariales del sistema están por lo general implicados en gran medida al especificar los cambios, resaltando la importancia de proporcionar al staff IT y al personal empresarial una vía para que se comuniquen de manera clara y efectiva con relación a los cambios. Un modelo de flujo de trabajo que expresa el propósito empresarial central de la aplicación, desprovisto de cualquier material técnico irrelevante, es una forma efectiva de lograr esta comunicación.

Estas aplicaciones también requieren flexibilidad dentro de la estructura del flujo de trabajo. En una aplicación IVR, por lo general el usuario está muy limitado, se mueve a través de un conjunto de menús estructurados de forma jerárquica. Sin embargo, también existen comandos de escape –por ejemplo, “regresar al menú principal” o “salir del subárbol”.

La aplicación de un centro de llamadas tendrá más flexibilidad que la aplicación IVR si se cambian las opciones ofrecidas al usuario

en respuesta al estado de una orden o en respuesta a la entrada de un cliente como por ejemplo dejando de lado las peticiones de ventas en el caso de que el usuario comience a reaccionar de forma negativa.

Este tipo de aplicaciones requiere soporte para la gama de organizaciones de los ítems de trabajo, combinando secuencias, bucles y condiciones con saltos desde un estado a otro y también el tipo de comportamiento orientado a los datos que se percibe en la colaboración para resolver el problema.

Norma y política

Tal como lo analizamos previamente, una forma en la que el enfoque del flujo de trabajo puede proporcionar valor es aislando el foco del cambio en una aplicación. Con frecuencia, este foco se encuentra sobre la forma en la que los ítems de trabajo se estructuran, pero en algunas aplicaciones, el foco del cambio reside sobre las expresiones asociadas a una estructura de cambio relativamente lenta.

Un ejemplo de este foco es un sistema de cotización de una política de seguros, donde un conjunto de cálculos en constante cambio se utiliza para orientar la toma de decisiones en el proceso de cotización. El requisito es para que el flujo de trabajo modele estas expresiones, lo que brinda dos beneficios claves. En primer lugar, los costos de implementación y evaluación son mucho más bajos que aquellos en los que por lo general se incurriría si las expresiones estuvieran escritas como códigos, ya que el modelo proporciona una fuerte zona protegida que restringe el alcance de cambios posibles. Segundo, los cambios podrían realizarlos personas que comprenden la importancia comercial de las expresiones pero que tal vez no posean la capacidad de comprender el código técnico en el cual las expresiones escritas como código deberían incrustarse inevitablemente.

El patrón Modelo-Vista-Controlador (MVC) por lo general se utiliza para conectar un UI a un modelo de objeto subyacente (Ver Figura 3). El *modelo* representa el comportamiento del sistema, independiente de cualquier representación UI particular. El *controlador* es la parte de la capa del UI que se utiliza para asignar eventos generados por el UI dentro de las llamadas del método requeridas para orientar el modelo. Por lo tanto, el UI en sí mismo no está viciado por ninguna suposición sobre el modelo subyacente.

Los flujos de trabajo considerados hasta el momento, vistos desde este punto de vista, entran todos en la categoría de modelos en el sentido MVC. Sin embargo, el controlador puede también ser visto como un flujo de trabajo. Los ítems de trabajo que organiza son métodos provistos por los objetos del modelo. El controlador también interactúa con el IU y el modelo por medio de contratos bien definidos. Un modelo de este tipo por lo general se denomina *flujo de página*.

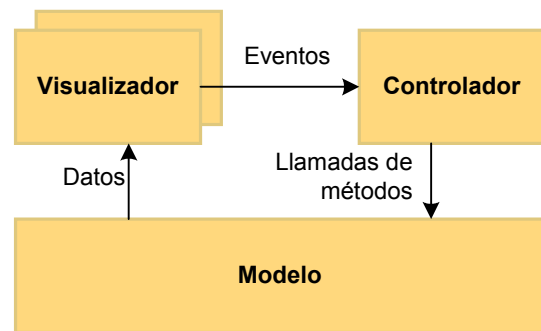
Al igual que con las operaciones con secuencias de comandos, el flujo de página no se implementaría hoy en día utilizando un producto de flujo de trabajo típico. Existen dos razones para considerar la construcción de un flujo de página utilizando una plataforma de flujo de trabajo. Primero, un modelo podría con facilidad representarse de forma visual, colaborando con los desarrolladores y analistas para expresar y comunicar el comportamiento requerido. Segundo, si el flujo de página cambia con frecuencia, entonces la abstracción del flujo de página como un modelo mejora de forma ágil.

Existen dos requisitos importantes si este problema se va a tratar utilizando una plataforma de flujo de trabajo. El tiempo de ejecución del flujo de trabajo debe ser ligero ya que un flujo de página podría ejecutarse dentro de una aplicación pequeña sobre un escritorio y los contratos admitidos deben incluir características del contrato basado en el evento de los UIs así como también los contratos objeto del método expuestos por el modelo.

Ahora analicemos un ejemplo de una aplicación de *evaluación de registro/repetición*. La intención de este ejemplo final es evaluar los límites de la aplicabilidad de la hipótesis de la plataforma de flujo de trabajo.

La aplicación aquí es una herramienta para evaluar la construcción de aplicaciones como un conjunto de servicios. La herramienta utiliza un mecanismo de intercepción para registrar

Figura 3: Aplicación MVC



todas las interacciones entre los servicios que ocurren durante la ejecución manual de un caso de evaluación para la aplicación. Este registro podría repetirse. Durante la repetición, los mensajes originados de forma externa son generados sin intervención manual, y los mensajes entre el conjunto de servicios que constituyen la aplicación se verifican por secuencia y contenido en comparación con el registro original.

El flujo de trabajo es el caso de evaluación, la organización de unidades de trabajo que constituyen los servicios que participan. El flujo de trabajo es activo, en el sentido que simula el comportamiento de los mensajes originados de forma externa, y es también pasivo, ya que monitorea las interacciones entre los servicios.

Una característica única de esta aplicación es que el flujo de trabajo es escrito, no por un desarrollador o por un usuario sino por un programa, como parte del acto o registro del caso de evaluación. La creación del modelo del flujo de trabajo debe ser totalmente programable. Existen también requisitos de actualización dinámica y expansión.

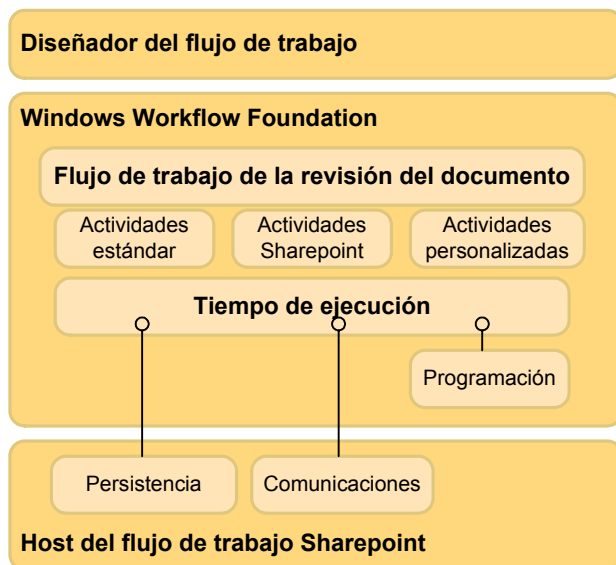
Se requiere expansión debido a que las semánticas estructurales son ricas. Por ejemplo, sólo porque dos mensajes llegaron a un servicio uno atrás del otro en el registro, no hay una implicación necesaria de que se debe preservar este orden en un registro. Si no existe una dependencia causal entre los mensajes, entonces una repetición que cambia el orden de los mensajes es correcta. Por lo tanto, la semántica de la secuencia en el modelo utilizado para registrar el caso de evaluación necesita incluir una noción de causalidad, que no necesariamente debe ser una característica del modelo del flujo de trabajo central de la secuencia.

Se requiere actualización dinámica porque la interacción humana con el modelo ocurre durante la repetición. Las discrepancias que se descubren durante la repetición entre el comportamiento observado y el registrado son direccionadas instantáneamente a un verificador. Si la discrepancia se debe a que el mensaje incluye una marca de hora que varía de ejecución en ejecución, entonces el verificador debería actualizar el modelo para marcar el campo "sin importancia". Si la discrepancia ocurre en una evaluación de regresión debido a que el software ha cambiado, entonces el verificador puede aprobar el cambio y actualizar la evaluación para esperar el nuevo comportamiento en todas las ejecuciones posteriores.

Valor de la plataforma de flujo de trabajo

Una plataforma de flujo de trabajo no posee por definición el conjunto total de características ofrecidas por los productos de flujo de trabajo típicos de hoy en día. Sino más bien, la plataforma de flujo de trabajo que se considera aquí se centra sobre la admisión del concepto de un flujo de trabajo como un modelo de organización de los ítems de trabajo. Hemos visto que la idea de una organización de ítems de trabajo es desde luego aplicable entre una gran variedad de aplicaciones, pero el hecho de que pueda utilizarse una plataforma de flujo de trabajo no significa que debe utilizarse.

Figura 4: Esquema de implementación de la revisión de un documento



Debemos plantearnos dos preguntas: ¿Qué valor adicional deriva del enfoque de la plataforma de flujo de trabajo? y ¿Es este enfoque práctico? Este valor del enfoque de la plataforma de flujo de trabajo debe provenir de la expresión de la organización del trabajo como un modelo, que analizaremos más adelante. Resumamos las características que debe mostrar una plataforma de flujo de trabajo efectiva y práctica.

Para demostrar la forma en la que un modelo varía de un código, el siguiente fragmento de códigos es un flujo de trabajo válido por la definición utilizada aquí, es decir, una organización de unidades de trabajo:

```
public void HandleLoanRequest (
    string customerId,
    Application app)
{
    if (CheckCredit(
        customerId, app.Amount))
    {
        MakeOffer (customerId, app);
    }
}
```

Y, en cierto sentido, es un modelo. Es posible descomponer este código y construir un árbol CodeDOM que lo represente.

Sin embargo, las semánticas del modelo resultante son demasiado generales como para ser opacas. Es posible decir que el código contiene llamadas de función, pero no es tan fácil distinguir una función que representa la llamada de un ítem de trabajo de una función que convierte números enteros en secuencias. Un modelo de flujo de trabajo distingue estas ideas de forma explícita. Por lo general, un elemento de modelo especializado se utiliza para representar la llamada de un ítem de trabajo y las funciones de conversión no se pueden expresar de forma directa en el modelo. Por lo tanto, un modelo de flujo de trabajo es aquél en el cual su gráfico se construye desde elementos que son significativos en el dominio del flujo de trabajo. La riqueza semántica de tal modelo puede explotarse de diversas formas.

Visualización. La representación visual del modelo –por lo general en forma de gráfico– es útil para los desarrolladores tanto en la creación como en el mantenimiento y también para los usuarios del flujo de trabajo quienes desean conocer la razón por la cual se les ha asignado una determinada tarea, o para el trabajador de operaciones IT quien desea comprender lo que debería realizar una aplicación que no se comporta como se espera.

Introspección. El modelo de flujo de trabajo es mucho más factible de acceso programático para una variedad de propósitos. Un ejemplo es el análisis estático para determinar las dependencias y flujo de trabajo entre un conjunto de flujos de trabajo en cooperación o utilizando el modelo para orientar una simulación que predice las cargas de trabajo que serán generadas por una nueva versión de un proceso.

Expresividad. La especialización del modelo de trabajo para el dominio de flujo de trabajo significa que los problemas característicos pueden expresarse de forma más rápida y compacta. Se trata de un lenguaje específico de dominio (DSL) especializado para soportar problemas característicos. Consideremos el proceso de revisión de un documento en el que tres votos positivos contra cinco revisiones implica que el documento es correcto, y cualquier otra revisión pendiente puede cancelarse. Este proceso es bastante difícil de codificar, pero un modelo de flujo de trabajo puede proporcionar construcciones listas para usar para tratar tales problemas.

Una explotación más semántica

Tal como lo hemos visto en el análisis de aplicaciones para operaciones con secuencias de comandos, ampliar el modelo de flujo de trabajo para especializar aún más el lenguaje modelo listo para usar es una técnica muy eficaz que proporciona valor adicional. Un ejemplo es la creación de un lenguaje hecho para los usuarios finales, al igual que en la revisión de un documento que se llevó a cabo utilizando una definición improvisada del proceso de revisión que analizamos anteriormente.

Ejecución. La especialización del modelo posibilita agregar soporte de tiempo de ejecución para problemas comunes. Un buen ejemplo es el estado de ejecución prologada. De las aplicaciones que hemos analizado aquí, se requiere la administración del estado de ejecución prolongada para el proceso de revisión de un documento, colaboración en la solución de problemas y aplicaciones de usuario guiado. El tiempo de ejecución de la plataforma de flujo de trabajo puede resolver inmediatamente problemas muy difíciles utilizando elementos del modelo de expresión simple para controlar una capacidad común y liberando al desarrollador para que se enfoque en el problema del negocio.

Monitoreo. La existencia de un modelo posibilita producir un flujo de eventos con una semántica significativa sin la necesidad de ningún esfuerzo adicional por parte del desarrollador. De las aplicaciones descritas aquí, este flujo de eventos es útil para la

“WF IMPLEMENTA LA IDEA DE UN FLUJO DE TRABAJO COMO UNA ORGANIZACIÓN DE ÍTEMS DE TRABAJO, EXTRAÍDOS DE LAS IDEAS RELACIONADAS CON LAS CUALES SE HAN ACOPLADO EN LOS PRODUCTOS DE FLUJO DE TRABAJO TRADICIONALES”.

revisión de un documento, la colaboración en resolución de problemas, evaluación de registro/repetición y aplicaciones de usuario guiado. El flujo de eventos puede utilizarse para monitorear instancias de flujo de trabajo o para construir vistas agregadas del estado de una gran cantidad de instancias de flujo de trabajo. La estandarización del flujo de eventos facilita mucho más la construcción de estas vistas agregadas entre los flujos de trabajo que se han desarrollado de forma independiente unos de otros.

Otra idea muy eficaz es la presentación de errores utilizando una semántica empresarial. Con frecuencia, una falla técnica como la omisión del envío de un mensaje, lleva a que se derive a un experto técnico ya que la importancia de la falla no es clara sin la investigación de un especialista. Si el error puede asignarse a un modelo de flujo de trabajo –para que sea claro que el error tiene que ver con la notificación de un cambio que no es importante, por ejemplo– entonces la derivación puede restringirse para los casos en los que sea necesaria.

Composición. Si una aplicación puede incluirse dentro de unidades de flujo de trabajo, entonces, estas unidades de trabajo, junto con sus interfaces bien conocidas, pueden ser reutilizadas por otros flujos de trabajo. Los flujos de trabajo en sí mismos también definen unidades de trabajo que a su vez pueden ser utilizadas por otros flujos de trabajo.

Personalización. Supongamos que ISV envía un flujo de trabajo, que ha sido personalizado por VAR y luego por un cliente. Reaplicar estas personalizaciones cuando ISV envía una nueva versión base constituye un problema de mantenimiento desafiante. El uso de un modelo compartido, bien conocido por el flujo de trabajo hace mucho más manejables las combinaciones triples resultantes. La personalización junto con la composición permite ecosistemas en los que las definiciones de trabajo y flujo se vuelven artefactos compartidos o comercializados.

Manipulación. Tal como hemos visto en los análisis de la revisión de un documento y la aplicación de evaluación de registro/repetición, existen con frecuencia requisitos para inventar o modificar flujos de trabajo sobre la marcha. Esta modificación no se puede realizar de forma segura si se requiere un cambio de código. Al utilizar un modelo se posibilita la manipulación dinámica que es controlable y general.

Estos beneficios constituyen una lista persuasiva, demuestran de forma clara que la descripción de una organización de ítems de trabajo como un modelo posee mucho para ofrecer.

Características de la plataforma

Debe existir soporte para conceptos estructurales básicos como secuencias, condiciones y bucles. Sin embargo, también es necesario que exista soporte para los enfoques orientados a los datos para poder tratar con organizaciones menos estructuradas que aparecen en aplicaciones tales como colaboración para la resolución de un problema y usuario guiado.

También es importante que se puedan incluir nuevos elementos de semántica para crear lenguajes especializados y ricos como composiciones que admitan flujos de datos en las operaciones con secuencias de comandos. Agregar nuevos elementos de semántica podría llegar al extremo de exigir la redefinición de esas ideas fundamentales como secuencias –por ejemplo, en la aplicación de evaluación registro/repetición.

El flujo de trabajo también debe poder comunicarse en una amplia variedad de formas. Los flujos de trabajo responden a eventos UI, orientan diferentes tipos de servicio (humanos, programáticos y otros flujos de trabajo), y admiten consultas sobre el estado actual de sus contratos –por ejemplo, al determinar las acciones disponibles para un actor en una aplicación de colaboración para la resolución de un problema.

Si la plataforma de flujo de trabajo se utilizará en todas las aplicaciones en las que agrega valor, como por ejemplo MVC, entonces debe ser ligera. De igual forma, es necesario tratar los requisitos de rendimiento y escala que implican aplicaciones como por ejemplo la revisión de un documento.

Además, el modelo de flujo de trabajo en sí mismo debe ser totalmente programable, lo que incluye la creación de modelos – como por ejemplo en la aplicación de registro/repetición– así como también debe ser un modelo de actualización dinámica para admitir cambios inesperados, como ocurre tanto en la revisión de documentos como en aplicaciones de registro/repetición.

Ahora analicemos la realización de estas características necesarias en el Windows Workflow Foundation (WF). Hasta ahora

hemos recapitulado de forma efectiva el pensamiento que orienta a los desarrolladores de WF. El WF como una realización de estos conceptos es el medio por el cual este valor puede traducirse en soluciones producidas.

WF implementa la idea de un flujo de trabajo como una organización de ítems de trabajo, extraídos de las ideas relacionadas con las cuales se han acoplado en los productos de flujo de trabajo tradicionales. Estas abstracciones constituyen tres categorías principales: diseño y visualización, hosting y semánticas.

Diseño y visualización. Un flujo de trabajo en WF representa un árbol de ítems de trabajo (denominadas actividades). Este árbol puede manipularse directamente como un modelo de objeto. Se proporciona un diseño pero no se exige su uso. Es posible crear

“EL OBJETIVO DEL TIEMPO DE EJECUCIÓN DE WF ES BRINDAR LAS CARACTERÍSTICAS REQUERIDAS POR CUALQUIER FLUJO DE TRABAJO, Y POR LO TANTO EVITAR LA NECESIDAD DE RE-IMPLEMENTARLAS UNA Y OTRA VEZ EN DIFERENTES APLICACIONES, PERO SIN COMPROMETER LA FLEXIBILIDAD DE LA ABSTRACCIÓN DEL FLUJO DE TRABAJO”.

nuevos diseños especializados para comunidades de usuarios en particular o para organizaciones particulares de ítems de trabajo. También es posible especializar el diseño provisto, el que se puede utilizar no sólo dentro de Visual Studio sino también desde adentro de una aplicación de hosting arbitraria.

Hosting. El tiempo de ejecución del WF es lo suficientemente ligero como para ser almacenado en un contexto de cliente como por ejemplo en el shell de una aplicación de cliente rico. También posee la suficiente capacidad como para escalar cuando está alojado en un servidor host, como el Sharepoint Server producido por Office 2007. Las necesidades del tiempo de ejecución de los servidores de WF se abstraen como interfaces de proveedor para servicios como la ejecución de sub-procesos, transacciones, persistencia y comunicaciones. Se proporcionan soluciones de proveedores útiles listas para usar, pero pueden substituirse si es necesario.

Semánticas. Los diferentes problemas responden a las diferentes semánticas de los modelos. WF admite tres estilos principales de flujo de trabajo listos para usar: orientado a datos, flujo y máquina de estados. El flujo es ideal para las aplicaciones en las que el flujo de trabajo está en control como el ejemplo de las operaciones con secuencias de comandos. La máquina de estados es el mejor estilo cuando el flujo de trabajo está orientado por eventos externos, como en aplicaciones de usuario guiado o MVC. Un enfoque orientado a los datos es el indicado para las aplicaciones en las que las acciones dependen del estado, como por ejemplo en la colaboración para la resolución de problemas.

Estas semánticas pueden ampliarse al construir actividades personalizadas y crear un vocabulario de dominio específico para que se utilice en cualquiera de estos estilos. Sin embargo, debido a que la estructura de un flujo de trabajo se expresa a sí misma como un conjunto de actividades, se puede utilizar el mismo enfoque para definir semánticas totalmente novedosas y nuevos estilos, en el caso que sea necesario.

Tiempo de ejecución normal de un flujo de trabajo

El objetivo del tiempo de ejecución de WF es brindar las características requeridas por cualquier flujo de trabajo, y por lo tanto evitar la necesidad de re-implementarlas una y otra vez en diferentes aplicaciones, pero sin comprometer la flexibilidad de la abstracción del flujo de trabajo. Estas características comunes pueden dividirse en cuatro categorías principales: programación de la actividad, estado de ejecución prolongado y transacciones, excepciones y compensación y comunicaciones. Analicemos cada una en detalle.

Programación de la actividad. El tiempo de ejecución de WF define un protocolo de actividades que implementan todos los ítems de trabajo. Este protocolo determina la actividad básica del ciclo de vida (iniciado, ejecución, finalizado) y los estados adicionales necesarios para tratar excepciones (defectuoso, cancelado y compensación). Esta definición permite que el tiempo de ejecución de WF brinde una programación del trabajo para todos los flujos de trabajo.

Estado de ejecución prolongado y transacciones. El tiempo de ejecución de WF admite la implementación de transacciones ACID. Estas transacciones son particularmente útiles para mantener la lógica entre el estado del flujo de trabajo y el estado externo como por ejemplo el estado del mensaje y la aplicación. Sin embargo, las transacciones ACID no son adecuadas para administrar un estado de ejecución prolongado debido a sus consecuencias de bloqueo y aplicación. El tiempo de ejecución de WF implementa un mecanismo de recuperación y punto de control más amplio para verificar el estado de ejecución prolongado. Desde este punto de vista, las transacciones ACID se convierten en unidades de ejecución dentro de un marco más amplio. El desarrollador no necesita realizar ningún esfuerzo para obtener el beneficio de admisión de WF para el estado de ejecución prolongado, ya que implica un comportamiento predeterminado. No obstante, en el caso de que sea necesario un control más detallado, se proporciona un conjunto de elementos de modelo simple para este propósito.

Excepciones y compensación. La bien conocida idea de excepciones *throw-try-catch* es admitida por el tiempo de ejecución de WF y se representa en el modelo de flujo de trabajo listo para usar. Sin embargo, el tiempo de ejecución de WF

“LAS TRANSACCIONES ACID SON PARTICULARMENTE ÚTILES PARA MANTENER LA LÓGICA ENTRE EL ESTADO DEL FLUJO DE TRABAJO Y EL ESTADO EXTERNO COMO POR EJEMPLO EL ESTADO DEL MENSAJE Y LA APLICACIÓN”.

también admite una vista más amplia del manejo de fallas que incluye la idea de compensación para lograr una finalización exitosa de las unidades transaccionales.

Comunicaciones. Como hemos visto, es necesario que los flujos de trabajo se comuniquen de varias formas, lo que se refleja en el WF, que admite la comunicación por medio del método .NET, interfaces de eventos e interfaces de servicios. El soporte para el Marco de Comunicación de Windows también estará disponible en el futuro. Por lo tanto, WF realiza, de hecho, el enfoque de plataforma-flujo de trabajo propuesto aquí.

La Figura 4 muestra la semántica de implementación de alto nivel de la aplicación de revisión de un documento y la manera en la que se integra todo lo anteriormente analizado. Una implementación utiliza Sharepoint como el host del flujo de trabajo. El tiempo de ejecución de WF utiliza el servicio de programación preestablecido con WF. Sin embargo, los servicios de comunicación y persistencia preestablecidos son reemplazados con implementaciones especializadas para el host de Sharepoint. El servicio de persistencia almacena el estado del flujo de trabajo de ejecución prolongada en la base de datos del Sharepoint, y el servicio de comunicación pone a disposición del flujo de trabajo las utilidades de interacción del usuario rico. De hecho, ambos servicios se brindan listos para usar en Microsoft Office 2007.

Para definir el flujo de trabajo en sí mismo en la revisión de un documento se utilizan tres tipos de actividades. Primero, se utilizan actividades WF listas para usar que proporcionan elementos estructurales como *If-Else* y *While*. Segundo, se utilizan actividades proporcionadas como parte de Office 2007 para acceder a los servicios de comunicación del usuario de Sharepoint. Tercero, se utilizan actividades personalizadas para implementar las semánticas específicas de la organización con el objeto de realizar el envío y la delegación de forma reutilizable y estándar. El diseñador WF se utiliza como un medio para definir el flujo de

trabajo y también para proporcionar representaciones esquemáticas del estado de una instancia de flujo de trabajo de revisión de un documento para el propietario del flujo de trabajo.

Enfrentar los problemas

En síntesis, la plataforma de flujo de trabajo admite la abstracción de ideas que posibilitan que los productos de flujo de trabajo sean una forma atractiva para hacer frente a los problemas de negocios. Sin embargo, no reemplaza a los productos de flujo de trabajo de la actualidad. Sino más bien, que los incluye en una superestructura y plataforma.

La plataforma de flujo de trabajo representa dos ideas claves: un flujo de trabajo es una *organización de unidades de trabajo* y un flujo de trabajo es un *modelo*, es decir, es una descripción que puede ser leída por una máquina además de un código. Estas ideas son de gran valor para una gran variedad de aplicaciones, tanto dentro como más allá del dominio de un problema tratado por productos de flujo de trabajo típicos. Esta plataforma de flujo de trabajo es mucho más útil si es ubicua y de bajo costo.

Los beneficios principales que brinda surgen de la expresión de una organización de ítems de trabajo como un modelo, lo que posee varias ventajas sobre la representación en código:

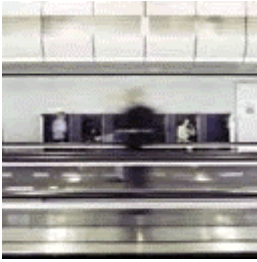
- **Transparencia.** Los objetivos empresariales de un sistema son claros permitiendo que los usuarios y el staff IT se comuniquen de forma efectiva respecto del comportamiento deseado y que el personal que se integra al proyecto comience a producir con rapidez.
- **Aislamiento del cambio.** Las áreas de aplicación que pueden cambiar con más probabilidad se expresan como flujo de trabajo más que como código. Al separar las partes de la aplicación que se mueven con rapidez, los cambios pueden realizarse de forma más segura.
- **Agilidad.** El resultado de todos estos beneficios es la agilidad del negocio. Si los usuarios del negocio pueden comprender el sistema, los desarrolladores pueden ponerse al día con rapidez, y los riesgos asociados con los cambios se minimizan. Por lo tanto, el sistema podría considerarse ágil.

Una plataforma de flujo de trabajo ampliamente útil debe poseer las siguientes características: debe definir un modelo de flujo de trabajo central como un estándar que sea extensible y totalmente programable en el tiempo de ejecución y diseño, debe poder comunicarse en una gran variedad de formas ricas, debe ser ligero e integrable y debe poder escalar y desempeñarse de forma adecuada en entornos de alto volumen. WF es un producto que muestra todas estas características. Como componente de WinFx y parte de la plataforma de Windows, WF también es ubicuo y de bajo costo.

El concepto de una plataforma de flujo de trabajo, tal como se la describe aquí, es un modelo configurable y adecuado que posee múltiples beneficios para una gran variedad de aplicaciones, y está totalmente realizado en WF. ●

Sobre el autor

David Green se incorporó a IBM como desarrollador para los laboratorios Hursley en 1977. Desde entonces ha avanzado a lo largo de toda la cadena de suministro de software, trabajando para una compañía de periódicos, American Express, Siemens Nixdorf y para un banco importante en el Reino Unido en una variedad de funciones técnicas, de pre y post venta. El tema presente a través de toda su experiencia es la construcción de aplicaciones para el mundo empresarial día a día, pensando que era mucho más difícil de lo que debería ser crear grandes soluciones empresariales, y tratando de crear enfoques y herramientas para enfrentar este desafío. David ha pasado los dos últimos años en Microsoft trabajando en Windows Workflow Foundation, lo que cree es una contribución importante para la armonía de quienes construyen aplicaciones.



La metáfora de “The Amazing Race”

Por Vignesh Swaminathan

Síntesis

La gestión de procesos de negocio (Business Process Management-BPM) se rumorea en la actualidad en oídos de todos, y varias empresas desarrollan el potencial empresarial de automatizar sus procesos de negocio. En la medida que el rumor se establece y el BPM se acerca a la meseta de la productividad, existen algunas consecuencias prácticas. Una de ellas es la necesidad de una mejor gestión de procesos de negocio de alto nivel que estén compuestos por procesos de negocio reutilizables y simples. La respuesta, interesantemente, no se encuentra totalmente en un modelado gráfico, sino más bien en la gestión de procesos de negocio de alto nivel a través de reglas de definición del proceso, externalizadas. Analizaremos la descripción, beneficios e implementación de las reglas de definición del negocio.

The Amazing Race (Carrera Asombrosa) de la cadena CBS es un Reality Show de la televisión de aire de los EEUU en el que participantes compiten en una carrera de resistencia alrededor del mundo yendo de ciudad en ciudad. Para ir de una ciudad a otra, utilizan medios de transporte comunes como un auto, tren o avión. Los equipos participantes deben encontrar una pista en cada ciudad para descubrir cuál es la próxima ciudad a la que deben viajar. Las pistas controlan la elección del próximo mapa de ruta a seguir; ni los participantes saben las pistas, ni el mapa en sí mismo contiene las pistas. Es decir, las pistas han sido externalizadas en el proceso de ir de una ciudad a otra.

Al utilizar esta analogía, todo el proceso de movilizarse alrededor del mundo (llamémoslo *proceso global*) se logra al combinar pequeños procesos de ir de una ciudad a otra (llamémoslos *procesos de ciudad*), y de este modo se crea el juego. Los procesos de ciudad que crean un proceso global en particular, son a su vez determinados por las pistas. Los participantes también deciden, por ejemplo, tomar un vuelo si no se puede acceder a la ciudad por ruta o tren. Estas son reglas internas simples que se aplican dentro del proceso de ciudad. Por otro lado, las pistas son totalmente externas al proceso de ciudad y controlan el proceso global.

Un cambio interesante para el juego sería que las pistas determinen el próximo proceso de ciudad en base a las características de los participantes, la ciudad en la que han estado previamente y otros parámetros relevantes. Este proceso podría llevar a una posibilidad más variada de ciudades a las que los participantes podrían viajar como próxima opción, cada vez que hayan llegado con éxito a una ciudad. Aunque este escenario se vuelve interesante para los participantes, ya no les conviene mucho a los creadores del juego. Ellos poseen ahora una matriz completa de posibles procesos de ciudad que pueden combinarse en el proceso global. Los procesos de ciudad

que se combinan son dinámicos y no están predeterminados (Ver Figura 1). Debido a que sólo las elecciones están predeterminadas y no el proceso de ciudad final, cuantos más parámetros formen la pista, más complejas serán las posibles elecciones para crear el proceso global.

La asombrosa carrera BPM de la empresa

Ahora, ¿Qué tiene que ver *The Amazing Race* con la Gestión de Procesos de Negocio (BPM) en la empresa? Las empresas poseen un enfoque muy similar a los procesos de negocio en su dominio. Cada empresa cuenta con una base de datos variada de los procesos de negocio que se pueden aplicar dentro de un subdominio particular en la empresa. Estos procesos de negocio de subdominio se modelan para lograr una tarea en particular o un flujo de trabajo dentro del contexto de este dominio. En la práctica, en la mayoría de los casos, las empresas comienzan a darse cuenta que los procesos de negocio que se aplican en el subdominio son reutilizables en un contexto más alto. Las empresas, entonces, terminan creando procesos de negocio para toda la empresa (alto nivel) reutilizando los procesos de negocio del subdominio (bajo nivel).

La madurez del enfoque para crear procesos de negocio de alto nivel tiene relación con la analogía de *The Amazing Race*. Las empresas se dan cuenta de la eficacia de combinar sus procesos de ciudad para realizar uno de sus procesos globales (Ver Figura 2). Entonces comienzan a combinar sus procesos de ciudad de formas más innovadoras utilizando reglas basadas en parámetros. En este punto también enfrentan el problema de manejar la complejidad que crean las posibilidades ricas del uso de reglas basadas en parámetros para controlar su proceso global.

Otro problema que presenta la analogía del juego es que el creador del juego posee un laberinto completo de opciones *if-then* para visualizar un único proceso global que podría constituir el episodio de la temporada en curso. Al utilizar este laberinto podría mantener el proceso global para la temporada en curso.

Imaginen una situación en la que debido a ciertas circunstancias imprevistas, como por ejemplo calamidades de la naturaleza, un estallido de guerra, terrorismo o algo que simplemente está fuera de control, los participantes del juego no pueden utilizar un proceso de ciudad en particular. En este caso, ellos deberían regresar y alterar el proceso global para asegurarse de que el proceso de ciudad problemático sea eliminado del proceso global o al menos posea alguna alternativa. Ya que el proceso global ahora está controlado por varios parámetros y reglas que se basan en estos parámetros, todo el modelo del proceso global debe examinarse rápidamente de forma manual para asegurar que no exista la posibilidad de alcanzar el proceso de ciudad problemático. Además, el show ya ha comenzado su temporada y el proceso global se está realizando. Este ejemplo, exige que, a pesar de

que el show ya ha comenzado, ¡ninguno de los equipos participantes termine en el medio de una ciudad en guerra!

El problema es aún más crítico en la empresa. El proceso global de la empresa (alto nivel) estará influenciado por los cambios que ocurran en el proceso de ciudad de la empresa (bajo nivel). Los motivos que impulsan los cambios pueden ser de igual forma imprevistos como ocurre en el juego. El negocio puede requerir que la empresa esté capacitada para cambiar de forma dinámica en un proceso de alto nivel para satisfacer las demandas nuevas e imprevistas del negocio (Ver Figura 3). El proceso global de la empresa también estaría en plena realización, y un cambio sería obligatorio aún cuando se está realizando el proceso para que de esta forma la empresa no termine en el medio de una situación similar a lo que sería un estallido de guerra. A diferencia del juego, las partes de la empresa no pueden simplemente cancelarse. ¡El show debe continuar!

Procesos de negocio de alto nivel

Tal como lo mencionamos anteriormente, los procesos globales son procesos de alto nivel y, como tales, son totalmente gestionados por usuarios no técnicos. Los creadores del juego son más expertos para gestionar un show que expertos en tecnología de programas de televisión. Este contexto también es cierto para la empresa en la que los directores ejecutivos gestionan el proceso de negocios de alto nivel, ya que conocen la forma de realizar las tareas para llevar a cabo sus negocios y no son expertos en la creación de modelos informáticos para su negocio. Por lo tanto, el primer aspecto más importante para gestionar procesos de negocio de alto nivel es la necesidad de abstraer al usuario de los tecnicismos.

La segunda necesidad reside en asegurar que el proceso global que se está llevando a cabo pueda alterarse dentro de

determinados límites. Esta flexibilidad representa un requisito importante para los procesos de negocio de alto nivel que están más sujetos a cambios en el proceso de lo que lo están los procesos de bajo nivel.

La tercera necesidad es proporcionar una interfaz clara, simple y fácil para que el usuario gestione el proceso global. Esta interfaz de gestión debe permitir la creación y mantenimiento de procesos de negocio de alto nivel. La misma interfaz también debe permitir que el usuario modifique de forma conveniente los procesos globales que se están llevando a cabo y debe posibilitar la búsqueda fácil de reglas basadas en parámetros que controlen el proceso de negocios de alto nivel. Esta búsqueda le permitiría al usuario encontrar rápidamente las reglas que desea modificar. La interfaz debe ser ligera por naturaleza para permitir que el usuario actualice de forma rápida y sin esfuerzo el proceso de negocios de alto nivel.

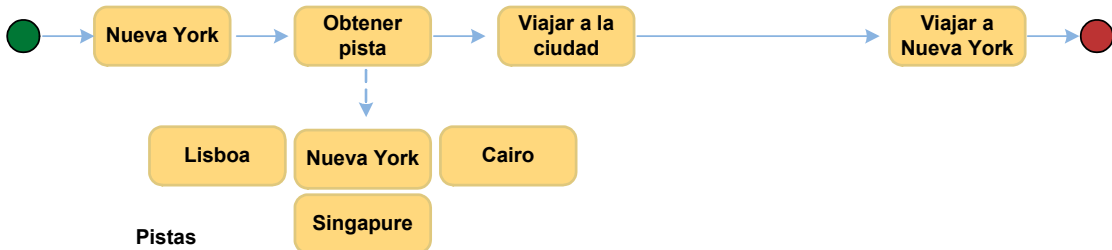
El requisito final es comprender que el proceso global está determinado por la combinación de una base de datos muy variada de los procesos de ciudad que se basan en reglas orientadas por parámetros. La cantidad de parámetros no está limitada en la práctica, pero puede exceder 30 parámetros en varios contextos de empresas en tiempo real. La primera vez que enfrentamos este problema práctico fue en un trabajo realizado para uno de nuestros clientes. El cliente, un banco importante con sede en Holanda, enfrentó este problema en su departamento bancario minorista. Contaban con un promedio de 5 parámetros definidos para controlar cada paso de la aplicación de un proceso de préstamo. Cada parámetro contenía en cualquier parte de 3 a 100 valores posibles que llevaban a procesos de negocio complejos, lo que hacía que se consideren difícil de gestionar a través de los medios convencionales.

Figura 1: Por el mundo

1 - Viaje por el mundo simple



2 - The Amazing Race (pistas estáticas)



3 - Pista inteligente de "The Amazing Race" (pistas basadas en parámetros)

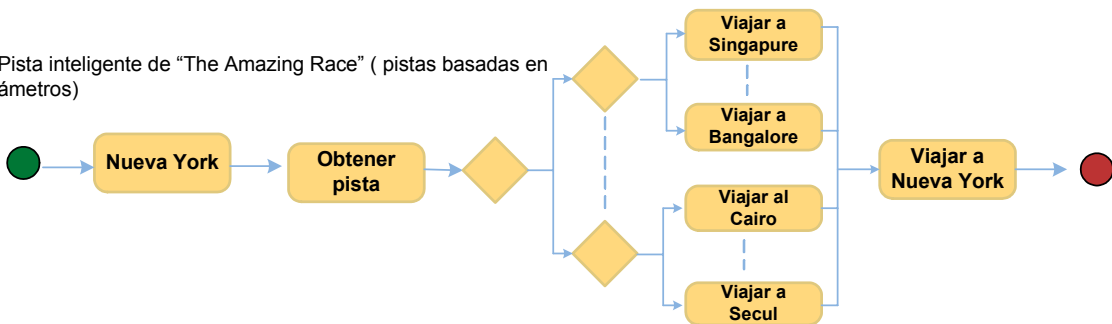
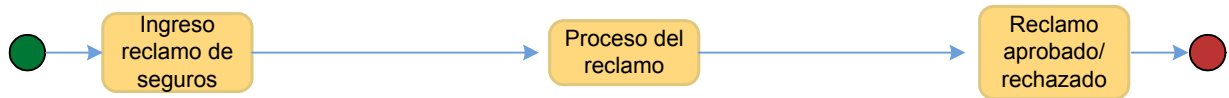
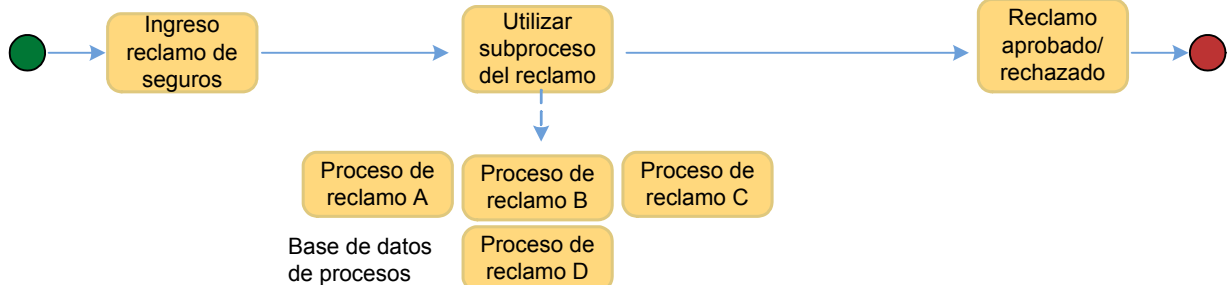


Figura 2: La empresa

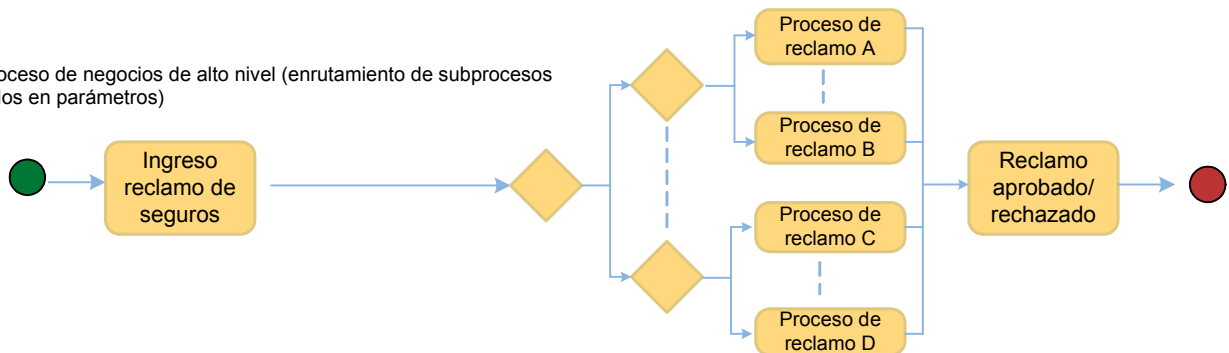
1 - Proceso de negocios simple (pasos redundantes)



2 - Procesos de negocios de alto nivel (llamadas a subprocesos estáticos)



3 - Proceso de negocios de alto nivel (enrutamiento de subprocesos basados en parámetros)



Para mostrar esta complejidad en acción, supongamos que *The Amazing Race* siempre cuenta con 5 equipos que compiten en cada nueva carrera (proceso global). Cada equipo, en cualquier etapa del juego, podría haber viajado a través de una lista de ciudades antes de viajar a la próxima ciudad. Por ejemplo, hay un total de diez ciudades por las cuales pueden competir los equipos, lo que representa diez posibilidades diferentes de la ciudad en la que el equipo está en ese momento. Ellos también podrían haber llegado a la ciudad en la que están utilizando alguno de los tres medios de transporte primarios como ruta, ferrovías o aire.

Utilizando esta analogía, la tarea de los creadores del juego sería componer el proceso global utilizando estos parámetros de cantidad de equipos, lista de ciudades previas y modo de transporte. Para lograr una combinación única de estos tres parámetros, es necesario combinar un proceso de ciudad particular en un proceso global. El proceso global debe asegurar que los participantes no utilicen el mismo medio de transporte más de una vez consecutiva, y también debe asegurar que finalmente los participantes lleguen a la ciudad en la que comenzaron para de este modo dar finalización a la carrera.

Aplicación de las reglas

El creador del juego también debe utilizar la cantidad de equipos para determinar ciertas condiciones especiales al azar y así asegurar que el juego cuente con un contenido más variado asegurando que los equipos viajen a las diferentes ciudades, en distinto orden y por diferentes medios de transporte. De este

modo, el próximo proceso de ciudad que se debe utilizar en cualquier etapa del proceso global estaría determinado por el equipo implicado, la ciudad en la que se encuentran y el último medio de transporte que utilizaron.

Esta combinación traduce aproximadamente 150 caminos de flujo diferente que pueden determinar el próximo proceso de ciudad que se utilizará en el proceso global. la cantidad de posibles caminos de flujo se determina de la siguiente manera: 5 equipos x 10 ciudades posibles x 3 medios de transporte que sólo pueden utilizar = 150 elecciones únicas para determinar el próximo

"EL PRIMER ASPECTO MÁS IMPORTANTE PARA GESTIONAR PROCESOS DE NEGOCIO DE ALTO NIVEL ES LA NECESIDAD DE ABSTRAER AL USUARIO DE LOS TECNISMOS".

proceso de ciudad (Ver Figura 4). Por ejemplo, el equipo número tres puede haber llegado a Nueva York por tren y luego determinar el próximo proceso como "a Londres por aire".

Esta complejidad también se traduce para las empresas. Tanto los dominios sobre seguros como financieros proporcionan diversos contextos que necesitan procesos de negocio de alto nivel controlados por reglas basadas en parámetros. Por ejemplo, un reclamo sobre seguros que debe ser investigado por una empresa de seguros global puede necesitar un reclamo de seguros, un proceso de negocios de alto nivel que seleccione un

Gestión de procesos de negocio automatizados

país específico, un proceso de investigación del reclamo basado en otros parámetros adicionales. Supongamos que el reclamo sobre seguros se clasifica en 5 grupos etarios posibles de 10 países diferentes de operación y pertenece a 3 grupos definidos por los límites de monto exigido: 5 posibles grupos etarios x 10 países diferentes de operación x 3 posibles límites de monto exigido = 150 opciones únicas para determinar qué proceso de investigación del reclamo debe utilizarse.

Por ejemplo, si el grupo etario es B (30-45), el país es India y el límite del monto exigido es mayor a \$ 150.000, entonces se utilizará el "Proceso de investigación de reclamo en India, Grupo C, Valor Alto".

Si volvemos al ejemplo financiero anterior que comprendía un banco con sede en Holanda, el cálculo matemático sería el siguiente: 5 posibles canales de interacción con el cliente x 3 posibles tipos de usuarios x 4 posibles segmentos de clientes x 100 posibles ofertas de productos x 7 posibles pasos para el proceso de alto nivel = 42.000 posibles implementaciones en la base de datos del proceso.

Un punto adicional para considerar: En la metáfora de *The Amazing Race* la cantidad de procesos de ciudad disponibles disminuye en cada repetición de la etapa del proceso del viaje a una ciudad. No obstante, en las empresas en tiempo real no es simplemente un paso del proceso que posee una gran variedad de posibles implementaciones del proceso, sino que todos y cada uno de los pasos del proceso de negocios de alto nivel puede tener múltiples implementaciones posibles. La metáfora se ha simplificado deliberadamente, y el incremento de la complejidad en cada nuevo paso que depende de reglas basadas en parámetros se muestra en la Figura 5.

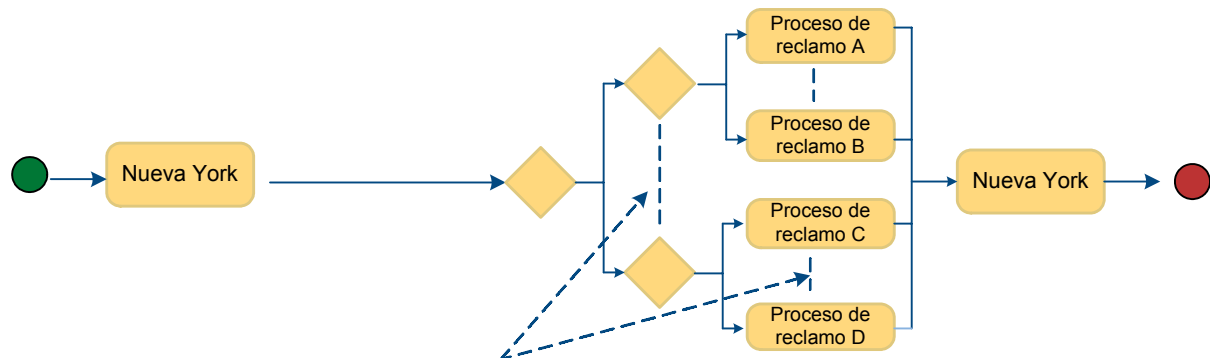
¿De qué forma podemos administrar estos contextos? ¿Aún deseamos utilizar modelos gráficos para administrar procesos de negocio de alto nivel?

La primera e inmediata noción que se le ocurre al usuario cuando tomamos los ejemplos de proceso global y procesos de ciudad es crear el proceso global con los procesos de ciudad como subprocesos que se utilizan dentro del proceso global. Esta noción no es sorprendente ya que el uso de subprocesos es la forma más común de combinar procesos simples en procesos más complejos. Utilizar subprocesos es una buena opción cuando tratamos de dividir un proceso de negocios en submodelos para una mejor reutilización. Sin embargo, no son aplicables para procesos de negocio de alto nivel que están controlados por reglas basadas en parámetros porque cuando se utilizan los subprocesos, las decisiones que controlan el flujo de los procesos de negocio de alto nivel están integradas y son internas al proceso.

Esta característica infringe la necesidad de flexibilidad establecida en los requisitos de los procesos de negocio de alto nivel. Tampoco resuelve el problema de gestión de decisiones ya que el usuario cuenta con opciones ineficaces para modelar 150 caminos de flujo distintos en un proceso de negocios de alto nivel y deberá cargar el enorme modelo gráfico resultante cada vez que sea necesario realizar un cambio en el proceso global. Además, este enfoque no está en línea con los preceptos de acoplamiento escaso orientados al servicio ya que los procesos finalmente serían expuestos como servicios.

Cada esfuerzo de optimización que se realice para desglosar los puntos de decisión dentro de los subprocesos también sería inútil debido a que el intercambio representa la visibilidad del proceso de negocios de alto nivel, y ahora el usuario enfrenta problemas adicionales ya que deberá profundizar para realizar las modificaciones. Aún si el usuario realizara una tarea paso a paso con las modificaciones, este enfoque no garantiza que se puedan realizar las modificaciones en el proceso global mientras está en ejecución. Por lo tanto, utilizar subprocesos en todos los aspectos no resuelve el problema de BPM de alto nivel.

Figura 3: La complejidad



La complejidad se encuentra en la cantidad de posibles opciones de la empresa en tiempo real, para seleccionar la implementación de un subproceso en particular. Esta complejidad aumenta con la cantidad de parámetros requeridos.

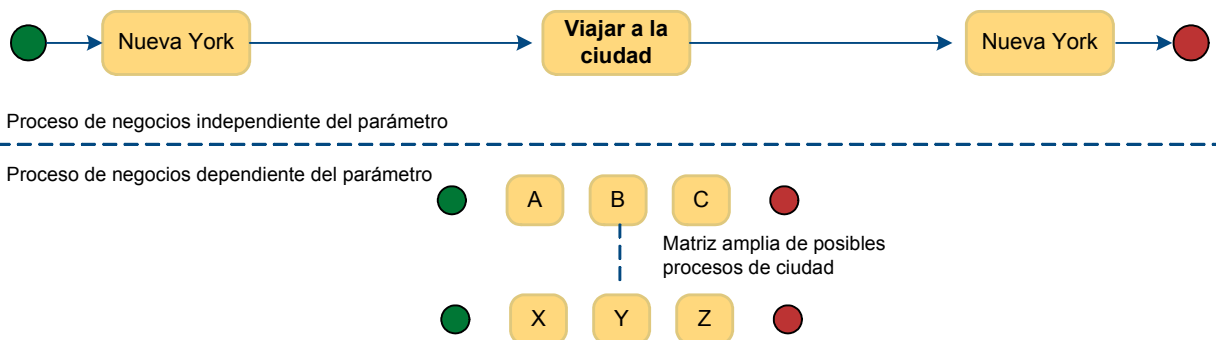
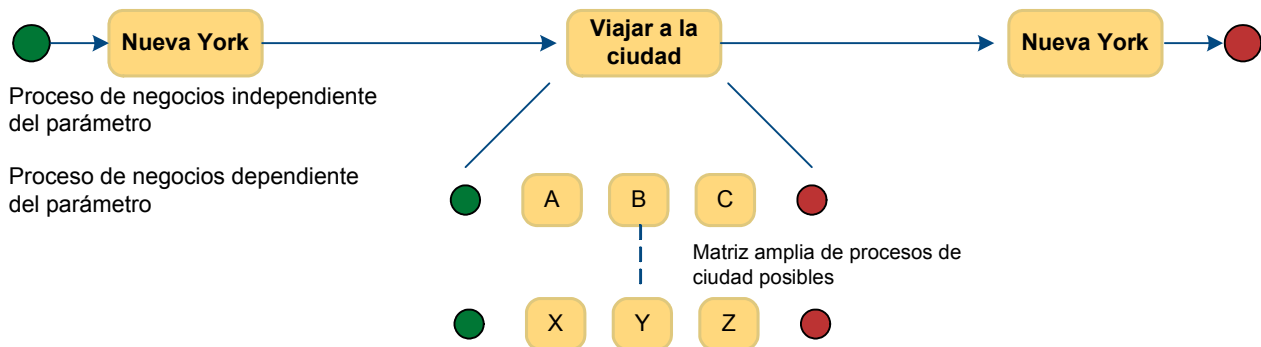


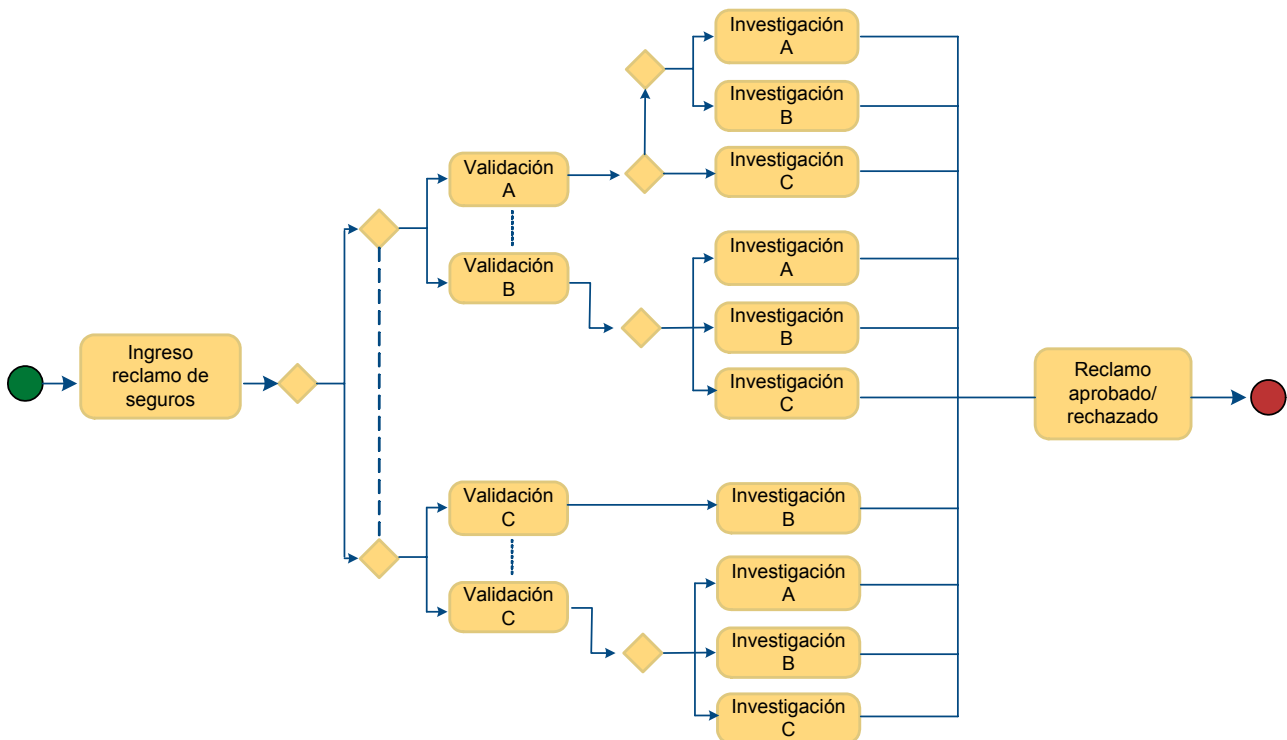
Figura 4: Determinación de la cantidad de caminos de flujo



Ejemplo: tres parámetros para una carrera por el mundo podrían ser: equipo, ciudad y transporte

Equipo		Ciudad		Transporte		Implementaciones únicas posibles
5	x	10	x	3	=	150!
Agregar un país más de operación y dos y dos valores más de la cantidad reclamada:						
5	x	11	x	5	=	275!

Figura 5: Un paso más



Condición externalizada

Una de las desventajas principales de la solución de subprocesos fue que las decisiones estaban incrustadas y estáticas dentro del proceso de negocios de alto nivel. Algunas herramientas BPM inteligentes brindan una solución que trata este problema específico. Estas herramientas proporcionan las capacidades para modelar los procesos de negocio que aplican la regla de decisión desde una base de datos de búsqueda o hacen referencia a un mensaje entrante durante la ejecución para la regla de decisión.

El enfoque aquí es lograr una base de datos para el almacenamiento de las reglas comerciales basados en parámetros, que controle el flujo del proceso de negocios de alto nivel (Ver Figura 6). En el modelo gráfico del proceso de negocios de alto nivel se realiza una simple llamada al servicio sobre el almacén de reglas para evaluar los datos del proceso de negocios y determinar el próximo paso lógico. Este enfoque reduce en gran medida la complejidad de los procesos de negocio de alto nivel en el sentido que ahora el modelo gráfico no posee los caminos de flujo pero sí posee los pasos lineales del proceso de negocios de alto nivel de una etapa a otra. Sin embargo, la solución aún no es lo suficientemente adecuada como para resolver el problema principal de manejabilidad.

Las desventajas de esta solución residen en que el almacén de reglas abstraer la lógica completa del proceso de negocios de alto nivel, lo que hace que el modelo sea menos transparente para los

usuarios de la empresa. El almacén de reglas y las posibilidades de interacción sobre los datos no son estándares y no garantizan que todos los detalles técnicos de las condiciones podrían ser abstraídos del usuario. Esto no nivela las herramientas estándar existentes ni los métodos disponibles. Por lo tanto, persiste el problema de una mejor solución.

Analicemos otra solución. Las reglas de definición del proceso (PDR) se gestionan de forma externa al modelo gráfico, de forma similar al enfoque de condición externalizada hasta el punto de mantener las reglas fuera del proceso de negocios de alto nivel. La diferencia reside en la forma en la que estas reglas externas se gestionan y cómo se controla el proceso de negocios de alto nivel.

“LOS PROCESOS DE NEGOCIO DE ALTO NIVEL SE CREAN AL COMBINAR VARIOS PROCESOS DE NEGOCIO DE BAJO NIVEL, FORMANDO, DE ESE MODO, DOS NIVELES DE BMP”.

La solución de una matriz PDR fomenta el motor de reglas como la capa de control y cambia el motor del proceso a una capa de ejecución, haciendo que el proceso de negocios de alto nivel fluya de forma más visible en las reglas más que en un modelo gráfico.

Con esta solución aún falta un enfoque estándar para modelar y mantener la base de datos PDR. El punto crítico aquí es que las abstracciones técnicas que proporciona una herramienta de

Figura 6: Una condición externalizada

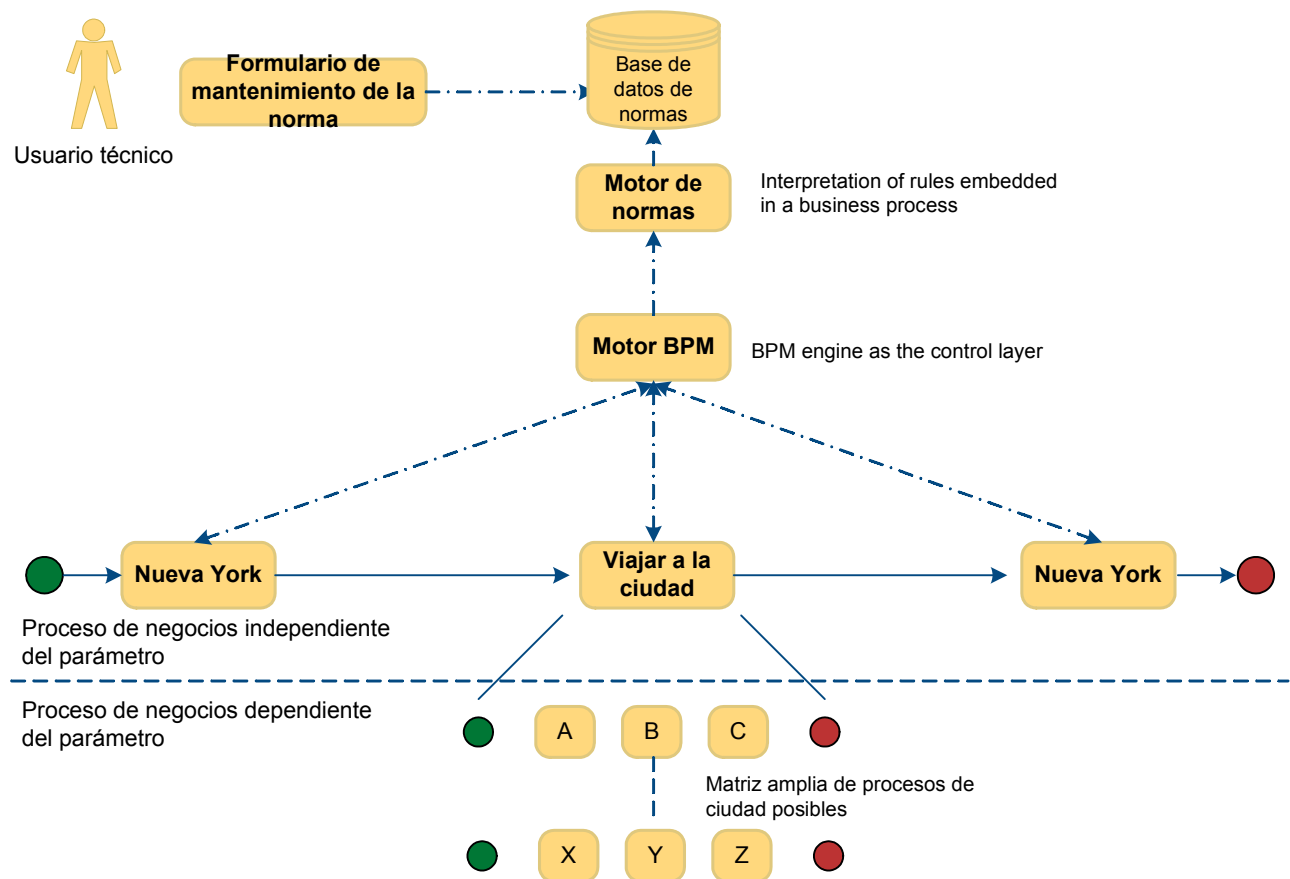
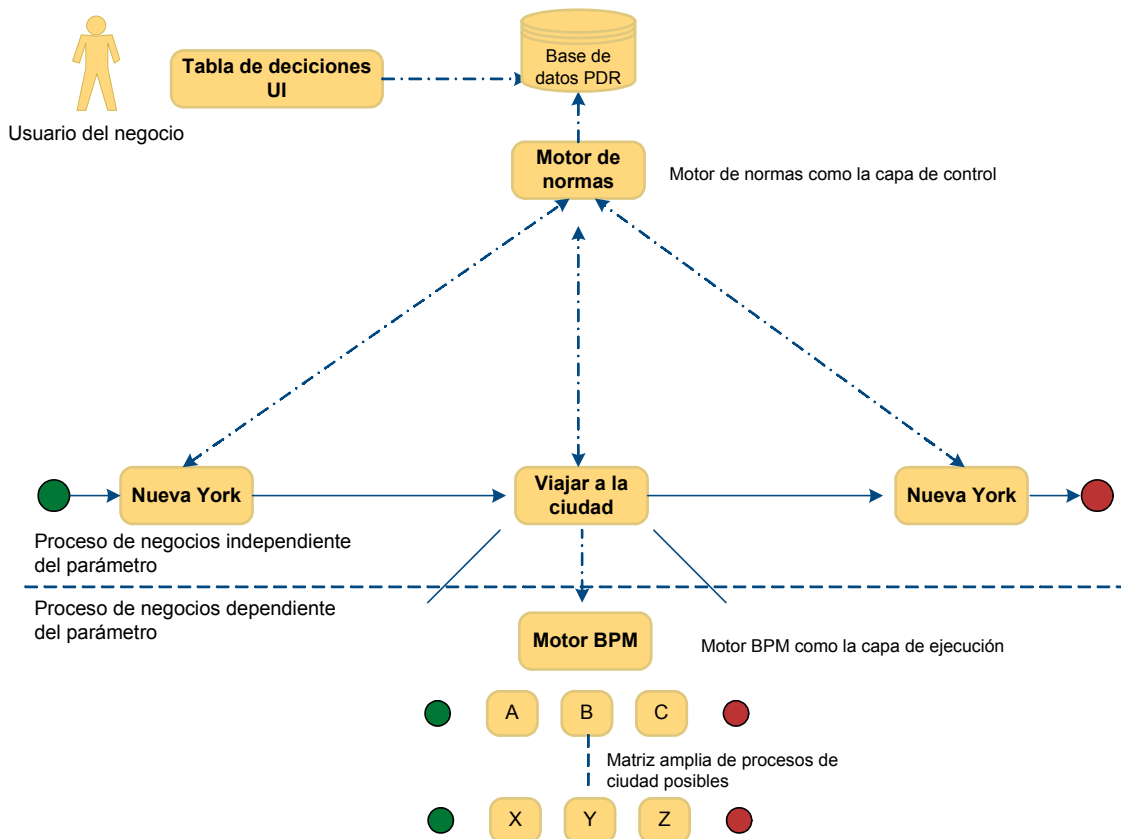


Figura 7: Una solución de matriz PDR



modelado del proceso de negocios gráfica deberían estar en perfecto estado para modelar y mantener la base de datos PDR. Debido a que la base de datos PDR es una base de datos de reglas, es necesario analizar las herramientas estándar disponibles en el mundo de reglas comerciales. Se destaca la tabla de decisiones ya que ofrece una solución adecuada para proporcionar una herramienta directa, liviana y sencilla.

La *tabla de decisión* es una herramienta estándar para modelar y gestionar la base de datos de reglas y conforma una excelente abstracción para los usuarios de la empresa que es simple de utilizar porque está basada en texto. Las tablas de decisión basadas en texto también son buenas candidatas para la búsqueda en la base de datos de reglas y permiten al usuario alterar de forma rápida las reglas importantes de definición del proceso.

Al utilizar esta herramienta se puede crear una solución que tenga un proceso de negocios de alto nivel gráfico que se simplifica al externalizar las reglas y se puede proporcionar una capa de servicio en función de la base de datos de reglas PDR (Ver Figura 7 y 8). De esta forma, la base de datos de reglas PDR se puede modelar y gestionar a través de la tabla de decisión. Denominamos a esta solución una matriz PDR, y los conceptos de proceso global y proceso de ciudad se pueden mostrar como dos niveles. El primer nivel representa el proceso de negocios de alto nivel de toda la empresa y el segundo nivel es un proceso de negocios de bajo nivel reutilizable.

El juego final

Hemos completado aquí un problema muy práctico con BPM en las empresas: la complejidad al definir y gestionar procesos de

negocios de alto nivel. Los procesos de negocio de alto nivel se crean al combinar varios procesos de negocio de bajo nivel, formando, de ese modo, dos niveles de BMP. Los procesos de negocio de alto nivel están marcados por la necesidad de reglas basadas en parámetros para seleccionar de forma dinámica un proceso de negocios de bajo nivel.

El uso de estas reglas basadas en parámetros para gestionar procesos de negocio de alto nivel, aumenta la complejidad del modelo gráfico múltiple con cada valor nuevo para cada parámetro. Este aumento de la complejidad hace que el modelo gráfico del proceso de negocios de alto nivel sea difícil de gestionar y poco práctico. La solución para este problema es utilizar un enfoque basado en las reglas para definir y gestionar los procesos de negocio de alto nivel.

Se puede definir y gestionar este enfoque utilizando una matriz PDR basada en una tabla de decisiones. Las tablas de decisiones son herramientas ligeras, no gráficas y fáciles de usar para crear y gestionar reglas. La matriz PDR abstrae la complejidad de gestionar un modelo gráfico de proceso de negocios de alto nivel. Por lo tanto, utilice una matriz PDR para gestionar BMP de dos niveles y vuelva a colocar al usuario comercial en el asiento del conductor de su empresa. ●

Sobre el autor

Vignesh Swaminathan es gerente de producto en CORDES R&D India (www.cordys.com), que proporciona un paquete de plataformas de aplicaciones de vanguardia que va más allá de los BPM y EAI básicos para resolver varios aspectos prácticos de la empresa. En los últimos cinco años ha trabajado en la organización de procesos de negocio, reglas comerciales, transformación de datos y otras tecnologías relacionadas. Se especializa en la integración de datos, procesos y personas. Comuníquese con Vignesh a vswamina@cordys.com y vigneshs@hotmail.com.

Recursos

CBC.com
www.cbs.com/primetime/amazing_race5



Explorar arquitecturas para el flujo de trabajo humano

Por Jesus Rodríguez y Javier Mariscal

Síntesis

Los sistemas de flujo de trabajo humano y algunos de los patrones más representativos de interacciones de procesos entre las personas y las actividades empresariales se dividen en dos componentes principales. El primero comprende los sistemas de flujo de trabajo humano y la interacción entre ellos en la medida que se implementan en plataformas de integración. El segundo componente representa los patrones de diseño del flujo de trabajo humano y la forma en la que son implementados utilizando interacciones entre los sistemas de flujo de trabajo humano. Este debate analizará con detención estos procesos.

La automatización total de los procesos de la empresa prácticamente no se puede lograr sin considerar la interacción de las personas, un factor que se vincula semánticamente a varios aspectos de la integración y automatización del proceso. La interacción de las personas está presente en algunos de los procesos de la empresa más comunes como por ejemplo la aprobación de órdenes y la gestión de recursos humanos. Estas interacciones humanas pueden variar desde un simple proceso de asignación de tareas a un proceso de negocios más complejo como la reasignación de tareas y la notificación.

Los esquemas para el flujo de trabajo humano están presentes en una variedad de servidores de integración popular de diferentes proveedores. Analizaremos de forma conceptual algunos de los componentes más importantes que están presentes en estas plataformas de flujo de trabajo humano y las interacciones entre ellos. También informaremos sobre algunos de los patrones de diseño de flujo de trabajo humano más comunes y la forma en la que se pueden implementar utilizando esos componentes.

Los sistemas de flujo de trabajo humano deben admitir la comunicación entre las personas y los sistemas. Para lograr esto, cada sistema de flujo de trabajo humano debe proporcionar funciones básicas como la asignación de tareas, la administración de identidad, notificaciones, seguimiento e interoperabilidad con los sistemas de gestión de procesos de negocio (BPM) (Ver Figura 1).

Los cuatro componentes principales de una arquitectura de flujo de trabajo humano son: un servicio de gestión de tareas, un servicio de seguimiento, un servicio de notificación y un servicio de identidad. Antes de analizar estos servicios en profundidad, es importante comprender la función de las tareas en los sistemas de flujo de trabajo humano.

Las tareas representan la unidad de comunicación principal entre los procesos de la empresa y las personas. Por lo general, una tarea se asigna a un usuario que necesita realizar alguna acción relacionada. Por ejemplo, un supervisor puede necesitar aprobar la solicitud de compra de algunos productos. Normalmente, un usuario debe realizar una serie de tareas que se agrupan de forma semántica. Es decir, el supervisor puede agrupar todos los ítems en un grupo llamado "aprobaciones del día" para llevarlo a cabo todos los días. Estos grupos se denominan listas de tareas.

Estados de la tarea

Durante el ciclo de vida de un sistema de flujo de trabajo humano, las tareas que se encuentran en la lista de tareas cambian constantemente de un estado a otro. Por ejemplo, la tarea del supervisor primero ingresa en estado de *pendiente*; cuando el supervisor acepta la tarea, el estado cambia a *reclamada*. Finalmente, cuando el supervisor aprueba la solicitud, la tarea ingresa su estado final de *finalizada*. Los estados se utilizan de forma conceptual para describir el ciclo de vida de las tareas. Algunos de los estados más comunes son *pendiente*, cuando se ha creado la tarea; *reclamada*, cuando un usuario ha reclamado la tarea y ha recibido sus datos de ingreso; *finalizada* cuando el usuario ha terminado la tarea y proporciona sus datos de salida; y *fallida*, cuando un usuario ha terminado la tarea y proporciona un mensaje de falla.

Las tareas, por lo general se asocian con esquemas de tiempo: caducidad, escalación, delegación y renovación. En nuestro ejemplo, la tarea de aprobación puede caducar si el supervisor no ejerce alguna acción sobre ella en un período de tiempo específico. Esta tarea que caducó puede posteriormente ser *escalada* hacia otro curso de acción o asignación. También, el supervisor puede decidir *delegar* la tarea a otra persona (un gerente, por ejemplo) para que actúe en su lugar. El gerente puede también decidir que *necesita* otro gerente para reunir puntos de vista adicionales. Si este segundo gerente no actúa sobre la tarea en un determinado período de tiempo, la tarea será *renovada* por otro período de tiempo.

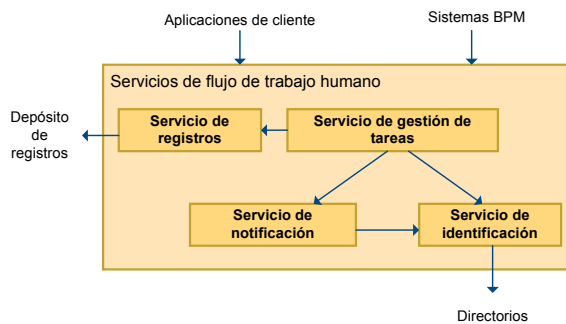
En otros contextos, las tareas se relacionan de forma semántica con las otras. La pregunta ¿Cuál es la próxima tarea? no siempre tendrá una respuesta trivial. En algunos casos, esta respuesta debe ser determinada por un tiempo de ejecución. Las tareas se pueden agrupar en secuencia dentro de una instancia de procesos de negocio para que el usuario sepa cuál es la próxima tarea que debe realizar antes de completar la tarea en curso.

En nuestro ejemplo, el supervisor debe aprobar 20 solicitudes en secuencia para completar el proceso del negocio. Cada vez que el supervisor completa una aprobación, el motor debe poder identificar la próxima tarea. Las *cadenas de tareas* representan un enfoque orientado por metadatos para describir la relación entre un conjunto de tareas dentro de un ámbito de un proceso de negocios. Las cadenas de tareas pueden agrupar las tareas de forma semántica para ayudar a que los usuarios obtengan funcionalidades como la gestión de una falla y la ejecución de una secuencia.

Las tareas que tienen dependencia sobre un proceso de negocios específico se conocen como tareas *en línea*. Las tareas que son totalmente independientes de un proceso de negocios particular son tareas *autónomas*. Las tareas en línea por lo general tienen acceso a los datos relacionados con un proceso de negocios y se almacenan en artefactos de procesos de negocio nativo como variables o mensajes. Por el contrario, las tareas autónomas interactúan con los procesos de negocio a través de una interfaz bien definida sin ninguna dependencia sobre los datos del proceso en sí mismo.

Servicios de flujo de trabajo

El componente de arquitectura que se ocupa de las tareas seccionales es el servicio de gestión de tareas. En un contexto

Figura 1: Componentes principales de una arquitectura de flujo de trabajo humano

típico, el *servicio de gestión de tareas* recibe una solicitud para crear una tarea, interactúa con el *servicio de identidad* para seleccionar todas las personas que son aptas para realizar la tarea, agrega la tarea a la lista de trabajos (tareas pendientes) asociadas con los usuarios seleccionados, y asigna políticas y plazos de tiempo específicos. Finalmente, un usuario decide trabajar con la tarea y la reclama. El usuario puede entonces trabajar con la tarea o solicitar más datos.

Un aspecto fundamental de los sistemas de flujo de trabajo humano es la capacidad de decidir el grupo de usuarios a quienes se les permite ejecutar una tarea. Este proceso de decisión sobre el usuario puede estar basado en interacciones con plataformas de gestión de identidad. En nuestro ejemplo, los servicios de flujo de trabajo humano necesitan identificar qué usuarios son aptos para aprobar la solicitud –en este caso el supervisor. Para lograr esta identificación, una plataforma de flujo de trabajo humano debe decidir el concepto de *supervisor* desde un conjunto de usuarios y funciones tradicionalmente almacenadas en un directorio de usuarios.

Se pueden establecer múltiples tipos de relaciones entre las personas y los procesos. Una de las más comunes es la forma en la que las personas interactúan con los procesos (funciones de las personas). Las personas de una empresa pueden agruparse en funciones que están relacionadas de forma semántica a alguna actividad de negocio, como por ejemplo administrador del proceso o propietario de la tarea. Otra relación común es la forma en la que los procesos identifican con qué persona interactuar (consultas y vínculos de personas). Dentro de un proceso de negocios, ciertos grupos de usuarios son importantes desde el punto de vista de la empresa. Los *vínculos de personas* se utilizan para representar los diferentes grupos de personas que participan en la ejecución de un proceso. Una *consulta* desde un directorio organizacional se utiliza para determinar los individuos asociados con un vínculo de personas y se refiere al vínculo de personas. En nuestro ejemplo, la función humana genérica, gerente financiero, podría ser habilitada por el supervisor de vínculos de personas, que está vinculado a la consulta “seleccionar autoridad de departamento, en la que el nombre del departamento es finanzas”.

El servicio de identidad está a cargo de las características relacionadas con el usuario como por ejemplo autenticación, autorización o determinación de personas. La información del usuario, por lo general se almacena en directorios organizacionales (por ejemplo, servicios de directorio Active Directory, un directorio LDAP o una base de datos relacionada). El servicio de identidad puede trabajar independientemente del directorio organizacional. En base al patrón adaptador, es posible extraer el acceso al directorio por medio del proveedor, que puede ejecutar las consultas para obtener la información desde el directorio. Este enfoque extrae las funcionalidades del flujo de trabajo humano desde el almacén del usuario.

En nuestro ejemplo, supongamos que el gerente quisiera reconstruir la ejecución del flujo de trabajo de aprobación de la solicitud para verificar deficiencias. El servicio de seguimiento mantiene un seguimiento de los cambios de estado relacionados con las tareas y cadenas de tareas. Este servicio debe proporcionar los fundamentos de la funcionalidad requerida para reconstruir los cambios del historial de tareas y realizar un análisis de las tareas.

También es necesario un servicio que notifique al supervisor por medio de un correo electrónico el momento en el que se crea la tarea de aprobación de la solicitud. El servicio de notificación administra los mecanismos de notificación para el usuario relacionado con los cambios de estado de las tareas.

Los cuatro servicios de tiempo de ejecución analizados aquí proporcionan una buena perspectiva general de las funcionalidades más comunes requeridas por los sistemas de flujo de trabajo humano. Si se combinan estos servicios, se pueden tratar algunos de los contextos más comunes de flujo de trabajo humano. Ahora veamos algunos de los patrones más comunes de flujo de trabajo humano.

Patrones de asignación de tareas

Los procesos orientados al flujo de trabajo han estado presentes en la industria por años. El conocimiento adquirido sirve como base para mejorar la construcción de sistemas de flujo de trabajo. Los patrones abstraen los sistemas de flujo de trabajo en diferentes niveles como por ejemplo la aprobación de tareas, la creación de tareas y la gestión del estado de las tareas. No es nuestra intención definir un catálogo de patrones de flujo de trabajo humano. Por el contrario, trataremos algunos de los patrones más comunes en los sistemas de flujo de trabajo humano desde el ángulo de la arquitectura que se ha definido anteriormente.

Comenzaremos con el análisis de un ejemplo de *flujo de trabajo de usuario simple*. Por ejemplo, un empleado, a través del portal del empleado, envía una solicitud de vacaciones. El portal inicia un proceso de negocio que incluye el modelado de una tarea del usuario utilizando un flujo de trabajo simple. La tarea es asignada al gerente de empleados. Cuando el gerente aprueba o rechaza la solicitud de vacaciones, se notifica al empleado por correo electrónico sobre la decisión del gerente.

Por lo tanto, esta solución combina los componentes de flujo de trabajo humano a través de cuatro interacciones. La tarea es configurada utilizando aplicaciones de cliente que interactúan con el servicio de gestión de tareas; se configura el ciclo de vida de la tarea o estados de la tarea. Como parte del proceso del negocio, la tarea se asigna a un grupo de usuarios utilizando el servicio de gestión de tareas. Uno de los usuarios reclama la tarea, y el servicio de gestión de tareas aplica la política adecuada para prevenir que otros usuarios actúen sobre la misma tarea. El proceso del negocio utiliza el servicio de gestión de tareas para obtener la actualización del estado de la tarea.

Ahora consideremos un ejemplo de un *flujo de trabajo secuencial*. El flujo de trabajo secuencial representa un contexto en el que una tarea debe ser aprobada de forma secuencial por un grupo de usuarios. Por ejemplo, cuando un sistema de aprobación de órdenes de compra procesa una orden de compra utilizando un proceso de negocios, un empleado que pertenece al grupo “supervisor” evalúa en principio la orden de compra. Una vez que el usuario inicial aprobó la orden de compra, el gerente de este usuario también la debe aprobar. Después que el gerente ha aprobado la orden de compra, ésta es reenviada a los departamentos de facturación y despacho. Esta solución posee cuatro interacciones. Interactúa con el servicio de gestión de órdenes para configurar las tareas y establecer las políticas adecuadas. Define la secuencia de usuarios que deben actuar sobre la tarea. Inicia la interacción de la tarea con el servicio de gestión de tareas. El primer usuario reclamará la tarea para comenzar a trabajar con ella, y una vez que ha finalizado, el servicio de gestión de tareas direccionará la tarea al próximo usuario del grupo.

Un patrón de *flujo de trabajo paralelo* representa el contexto en el cual una tarea debe ser aprobada por varios usuarios al mismo tiempo. Cada persona que aprueba puede agregar comentarios y anexos que son independientes de los otros. Por ejemplo, un proceso de contratación se utiliza para contratar nuevos empleados. Cada entrevistador vota a favor o en contra de un candidato. Si el 75 por ciento de los votos es favorable, se contrata al candidato. De lo contrario, el candidato es rechazado. El proceso se modela utilizando el flujo de trabajo paralelo, en el que cada entrevistador puede votar de forma independiente de los otros entrevistadores. Esta solución posee cinco interacciones para su implementación. Interactúa con el servicio de gestión de tareas para configurar las tareas y establecer las políticas adecuadas. Define la secuencia de usuarios que deben actuar sobre la tarea.

Arquitecturas para el flujo de trabajo humano

Inicia la interacción de la tarea con el servicio de gestión de tareas. El servicio de gestión de tareas direcciona la tarea a todos los usuarios. El servicio de gestión de tareas completará la tarea sólo cuando todos los usuarios hayan finalizado su actuación.

Asignación de tareas utilizando políticas

Ahora analicemos el uso de políticas para la asignación de tareas para lo cual es necesario asignar tareas a un usuario siguiendo reglas específicas. Por ejemplo, el procedimiento de un bypass para el corazón se asigna a un cirujano que tenga asignadas la menor cantidad de operaciones. La implementación de esta solución combina los componentes de flujo de trabajo humano por medio de cinco interacciones. Se interactúa con el servicio de gestión de tareas para configurar las tareas y establecer las políticas adecuadas. Se configura la política de la tarea como parte de la definición de la tarea –por ejemplo, se selecciona de un grupo de usuarios disponibles el que tenga la menor cantidad de tareas asignadas. Como parte del proceso de negocios, se asigna la tarea a un grupo de usuarios utilizando el sistema de gestión de tareas. Comienza la interacción de tareas con el servicio de gestión de tareas. El servicio de gestión de tareas ejecutará las políticas de la tarea entre los usuarios potenciales que pueden actuar sobre la tarea y, a su vez, se selecciona un usuario que se adapte a los criterios de la política y se le asignará la tarea a este usuario.

En el ejemplo de *flujo de trabajo de usuario único con escalación* se puede asignar una tarea a usuarios múltiples, pero sólo un usuario puede actuar sobre la tarea. Si la tarea caduca, el gerente del usuario debe actuar sobre la tarea. Por ejemplo, el proceso de solicitud de un servicio de mesa de ayuda permite a los usuarios presentar tickets de solicitud de un servicio de mesa de ayuda. Si la persona que recibe el ticket no actúa sobre éste dentro de un período de tiempo específico, el ticket se deriva automáticamente al gerente del usuario. Si nadie toma acción sobre el ticket dentro de un período de tiempo predefinido, el ticket se deriva automáticamente tres veces hasta llegar al director general de la compañía. Si el director general tampoco actúa sobre el ticket, éste caduca.

La implementación de esta solución combina componentes de flujo de trabajo humano a través de cinco interacciones. La tarea se configura utilizando aplicaciones de cliente que interactúan con el servicio de gestión de tareas; en este paso se configuran tareas como el ciclo de vida de la tarea, la política de escalación o los estados de la tarea. Como parte del proceso de negocios, se asigna la tarea a un grupo de usuarios utilizando el servicio de gestión de tareas. Un usuario reclama la tarea y el servicio de gestión de tareas aplica la política adecuada para prevenir que otros usuarios actúen sobre la misma tarea. Si la tarea caduca, el servicio de gestión de tareas la cancela.

En un ejemplo de *flujo de trabajo de usuario con delegación* un usuario que reclama la actuación sobre una tarea puede reasignarla a otro usuario. Por ejemplo, antes de tomarse una licencia, el jefe de contaduría transfiere todas sus tareas pendientes al asistente de contaduría. Para la implementación de esta solución se combinan los componentes de flujo de trabajo humano por medio de cuatro interacciones. La tarea se configura utilizando aplicaciones de cliente que interactúan con el servicio de gestión de tareas; se configuran propiedades tales como el ciclo de vida de la tarea, la política de escalación o los estados de la tarea. Como parte del proceso de negocios, se asigna la tarea a un grupo de usuarios utilizando el servicio de gestión de tareas. Un usuario reclama la tarea y el servicio de gestión de tareas aplica la política adecuada para prevenir que otros usuarios actúen sobre la misma tarea. Utilizando las aplicaciones de cliente apropiadas, el usuario inspecciona la lista de personas posibles para actuar sobre la tarea. El usuario reasigna la tarea a una de esas personas y reajusta las propiedades de la tarea.

Recursos

The Workflow Management Coalition
www.wfmc.org
OMG Business Process Initiative
www.bpmi.org

En el ejemplo de *patrón de tareas encadenadas* es necesario que una tarea se inicie en base al estado de finalización de otra tarea. Por ejemplo, el próximo ítem o ítems de trabajo comienzan en un proceso de coordinación de rescate de emergencia inmediatamente después que se ha completado la tarea anterior. Para la implementación de esta solución se combinan los componentes de flujo de trabajo humano por medio de cuatro interacciones. La tarea se configura utilizando las aplicaciones de cliente que interactúan con el servicio de gestión de tareas; se configuran propiedades tales como el ciclo de vida de la tarea o los estados. Se configuran los metadatos de la cadena, indicando que la segunda tarea se activará una vez que se haya completado la primera. La cadena de tareas se inicia a través del servicio de gestión de tareas, causando el inicio de la primera tarea. Cuando la primera tarea finaliza, el servicio de gestión de tareas utilizará los metadatos de la cadena de tareas para comenzar la segunda tarea.

Integración del proceso de negocios

Las soluciones de flujo de trabajo humano pueden diseñarse utilizando una combinación de servicios tales como la gestión de tareas, el seguimiento, la notificación y las aplicaciones de cliente. Hemos analizado la mayoría de los componentes más comunes presentes en arquitecturas de flujo de trabajo humano y su implementación en una serie de interacciones. Algunas de las más complejas interacciones de los sistemas con las personas pueden modelarse utilizando un conjunto de patrones de interacción entre personas que a su vez se implementan utilizando componentes de flujo de trabajo humano. Estos patrones y componentes básicos de arquitecturas de flujo de trabajo humano representan aspectos claves en la gama de integración de procesos de negocio, y al ser implementadas crean una solución eficaz de flujo de trabajo humano.

Hemos presentado aquí estos conceptos y contextos con el fin de ayudarlo a comprender mejor los componentes que por lo general se encuentran en plataformas de flujo de trabajo humano. Para obtener más información acerca de los conceptos de flujo de trabajo humano, consulte los recursos que se enumeran. También, consulte sobre las arquitecturas del motor de flujo de trabajo con alguno de los principales proveedores de servidores de integración.

Queremos agradecer a Kirsti Elliot, Ben Elliott y al equipo de estrategia de arquitectura de Microsoft por los comentarios y correcciones que realizaron en este artículo. •

Sobre los autores

Jesus Rodríguez es jefe arquitecto de software para Two Connect Inc. (www.twoconnect.com), un Gold Partner de Microsoft con sede en Miami, Florida. También es BizTalk Server MVP de Microsoft. La gran experiencia de Jesus respecto de la integración del proceso de negocios y el flujo de trabajo humano ha derivado de implementaciones múltiples de sistemas de acoplamiento escaso que se encuentran en los principios de SOA. Es colaborador activo de las comunidades .NET y J2EE, concentrándose en los aspectos de interoperabilidad de estas dos plataformas. Sus colaboraciones incluyen diversos artículos para diferentes publicaciones como MSDN, sesiones en conferencias de Microsoft como Teched y Webcast sobre diferentes tecnologías de Microsoft. Es un blogger prolífico sobre todos los temas relacionados con la integración y posee una verdadera pasión por la tecnología. Contacte a Jesus a jrodriguez@twoconnect.com o a través de su blog en <http://weblogs.asp.net/gsusx>

Javier Mariscal es presidente de Two Connect Inc. Javier ha dedicado una gran parte de sus 16 años de carrera profesional al diseño e implementación de soluciones de integración de aplicaciones y datos, en particular aquellas que implican la fusión de entornos AS/400 y grandes sistemas con la plataforma WinTel. Posee una verdadera pasión por soluciones de flujo de trabajo y automatización de procesos de negocio que implementan la técnica de desarrollo de aplicaciones orientadas al servicio (SODA) para el mantenimiento y desarrollo de aplicaciones distribuidas de forma global. Javier mantiene un trabajo bastante activo sobre tales soluciones para Fortune 1000 y empresas del mercado medio superior en todo el mundo. Contacte a Javier a Javier@twoconnect.com.



El Flujo de trabajo en la integración de aplicaciones

Por Kevin Francis

Síntesis

Uno de los mayores desafíos que enfrentan los arquitectos en la actualidad es la implementación de aplicaciones. Analicemos una técnica para la integración de aplicaciones que deja a un lado los enfoques comunes de integración específica y se dirige hacia una estructura consistente. Se describen los requisitos para una integración exitosa junto con la presentación de un enfoque de arquitectura para cumplir con estos requisitos utilizando herramientas tales como tecnologías de flujo de trabajo.

La integración de aplicaciones se ha vuelto cada vez más común y la creciente disponibilidad de herramientas y estándares (como los estándares de servicios Web WS-I) y la arquitectura orientada al servicio (SOA) parecen prometer una integración más fácil. Existen varios artículos que fomentan la simplicidad de vincular aplicaciones utilizando servicios Web o aún SOA. Hoy en día, comúnmente se utilizan dos enfoques para integrar aplicaciones: integración de un bus de servicios e integración punto a punto (Ver Figura 1).

En el contexto punto a punto se crean vínculos directos entre aplicaciones por medio de un vínculo directo a una Interfaz para Programas de Aplicación (API), un Protocolo de Transferencia de Archivos (FTP) o interfaces por lotes. La transformación (traducción) de la información puede realizarse mientras se transfieren los datos entre el vínculo. Por lo general, las interfaces punto a punto se implementan sin utilizar un producto de integración y la traducción de la información se realiza utilizando un código en el punto de integración en uno o ambos extremos de la interfaz.

La integración de un bus de servicios utiliza una solución de tecnología para proporcionar un bus sobre el cual las aplicaciones

pueden dejar mensajes para que el bus en sí mismo gestione el enrutamiento de mensajes entre las aplicaciones. Por lo general, el bus también gestiona la transformación de los formatos del mensaje entre las aplicaciones.

En la medida que las empresas se esfuerzan por brindar una mayor variedad de servicios online, intentan integrar líneas de productos y hacer más efectivas las experiencias del centro de

Tabla 1: Requisitos para una capa de integración

Capa	Requisito	Descripción
Datos	Conectividad	Conectividad básica en la que las aplicaciones pueden comunicarse unas con otras.
	Transformación	Traducción del formato de datos y por lo tanto entre las aplicaciones.
Información	Agregado de datos	Vista agregada de datos entre sistemas múltiples.
	Reglas comerciales	Desarrollo de reglas comerciales entre sistemas múltiples.
	Gestión de la transacción	Capacidad para realizar transacciones ACID entre múltiples sistemas.
	Modelo de información	Modelo de datos consistente entre todos los sistemas en el que se logra una comprensión común de la estructura y las entidades de datos.
	Gestión de datos de referencia	Gestión de datos de referencia utilizados comúnmente desde los múltiples sistemas en una ubicación única.
	Gestión de sesión	Gestión de información de sesión entre interacciones y entre sistemas.
	Instrumentación	Punto común para el registro de la información operativa.
	Gestión errores	Enfoque sistemático para la gestión de errores desde un único conjunto de normas.
	Gestión de configuración	Capacidad para configurar la operación del tiempo de ejecución de los sistemas completos, configura su comunicación con los diversos sistemas que componen el entorno e implementa nuevas versiones de los diferentes componentes.
	Flujo de trabajo	Procesos de flujo de trabajo que pasan por los sistemas múltiples.
Proceso	Reglas comerciales complejas	Reglas comerciales reutilizables, compartidas que pasan por los sistemas múltiples.
	Modelado de procesos de negocio	Modelado de procesos de negocio entre los sistemas para la optimización e integración.
	Monitoreo de la actividad comercial	Monitoreo de la rapidez y efectividad de los procesos de negocio integrales para la optimización y seguimiento del tema.

Figura 1: Integración del bus de servicio e integración punto a punto

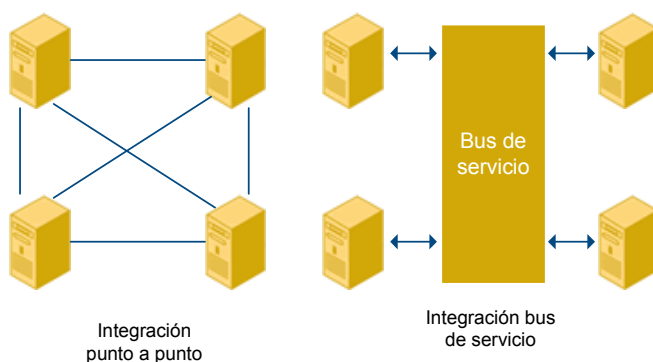
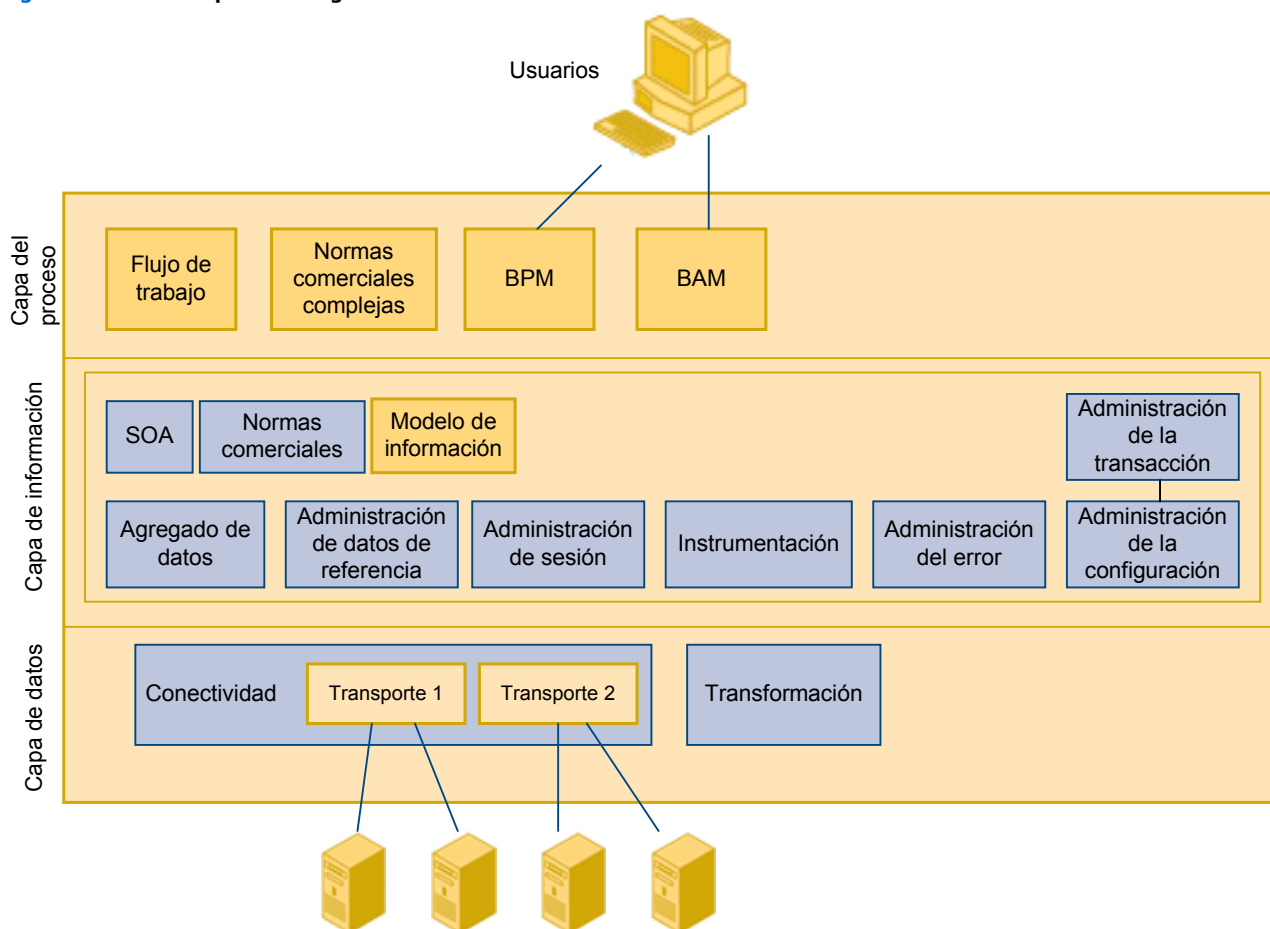


Figura 2: Las tres capas de integración



llamadas, una solución de integración requiere sistemas con más infraestructura informática. Por ejemplo, no es poco común para los operadores del centro de llamadas que cambien entre más de 10 aplicaciones cuando atienden las llamadas de los clientes. El reemplazo de este tipo de contexto en el que el usuario integra la aplicación de forma eficaz al copiar y pegar o al volver a escribir información en diversas aplicaciones para completar una transacción, es un orientador común para la integración de aplicaciones.

Eje central de la arquitectura de la empresa

Existen una cantidad de métodos que pueden utilizarse para integrar aplicaciones detrás de una aplicación de un centro de llamadas o un sitio Web, y la solución más común en la primera etapa es construir una cantidad de interfaces punto a punto o de bus de servicios.

Sin embargo ¿Qué sucede cuando una solución de un centro de llamadas y una aplicación de Web necesitan acceder a la misma información? El contexto ideal sería reutilizar las interfaces ¿Verdad? Bien, en la mayoría de los casos en los que los grupos de desarrollo (y los arquitectos) no están conectados, éste no es un enfoque común, dadas las complejidades que se suscitan al tomar una interfaz existente con su propio conjunto de códigos complejos y permitir que sea llamada por algo más. El problema aumenta de forma exponencial con la cantidad de interfaces que deben existir entre los sistemas y esto es relativo al tamaño de la empresa; las experiencias de empresas más grandes es sin duda peor que las experiencias de empresas más pequeñas.

Si no se implementa una solución integral bien diseñada da como resultado códigos duplicados en toda la empresa, enfoques de arquitectura incompatibles en cada sistema, y la inhabilidad

"SI NO SE IMPLEMENTA UNA SOLUCIÓN INTEGRAL BIEN DISEÑADA DA COMO RESULTADO CÓDIGOS DUPLICADOS EN TODA LA EMPRESA".

para responder a las necesidades de la empresa de forma inmediata. Estos problemas representan un producto secundario de un enfoque centrado en un proyecto para una arquitectura de solución.

SOA es comúnmente presagiada en la actualidad como la solución para la integración de problemas entre aplicaciones, pero se requieren una cantidad de características adicionales para una solución de integración verdaderamente eficaz. Estas características están enumeradas en la Tabla 1 y se agrupan en tres capas de integración:

- **Integración de datos** – La capa más básica, la integración de datos por lo general se logra aún en los contextos de integración más básicos. En esta capa, los datos se trasladan entre las aplicaciones transformándose para permitir que los datos puedan ser traducidos entre las aplicaciones.
- **Integración de la información** – En esta segunda capa, se agregan datos y llamadas a las aplicaciones para permitir que las llamadas únicas accedan a múltiples aplicaciones, teniendo

Tabla 2: Requisitos técnicos para la capa de datos

Requisito	Descripción
Conectividad	Conectividad es la capacidad de transferir información utilizando una variedad de protocolos o métodos e incluye la provisión de una interfaz de servicios Web. También incluye protocolos específicos necesarios para contextos específicos, los que cambiarán de una empresa a otra. Por ejemplo, varias empresas que conservan grandes sistemas para los sistemas de negocios centrales podrían requerir una conectividad IBM WebSphere MQ y/o SNA utilizando un Microsoft Host Integration Server (HIS). Otros sistemas pueden requerir interfaces COM+ o aún HTTP, raw sockets o FTP. Todos los protocolos deben ejecutarse a través de una interfaz común, con una implementación específica conectable para contextos específicos. Todos los requisitos técnicos específicos de los métodos o protocolos particulares deben ser tratados por la implementación y los sistemas externos no deberían necesitar contener ninguna lógica para tratar casos especiales.
Transformación	La transformación representa la transformación de datos de una estructura hacia otra en base a las reglas. Una vez más, el motor de transformación debe estar contenido dentro de la capa de integración.

Figura 3: Requisitos técnicos para la capa de información

Requisito	Descripción
Agregado de datos	La provisión de servicios que contienen llamadas múltiples traen con ellos agregados de datos. Es importante considerar el agregado de datos como un requisito por separado ya que aporta un requisito para una mayor atención sobre el diseño de los datos y el uso de un diseño de datos cuidadoso y reglas comerciales para permitir que los datos sean agregados más que simplemente acumulados.
Reglas comerciales	Es necesario prestar mucha atención al desarrollo de reglas comerciales en un contexto de integración para de este modo asegurar que sólo aquellas reglas que forman parte de cada aplicación se encuentren dentro de la aplicación, y aquellas reglas que están relacionadas con la integración de la aplicación sean encapsuladas dentro de la capa de integración. Este enfoque brinda la mayor oportunidad de reutilización, simplifica el mantenimiento de las aplicaciones y proporciona un punto único y centralizado para el desarrollo y ejecución de las reglas.
Gestión de la transacción	La gestión de la transacción es un requisito tanto necesario como complejo para la integración de la aplicación. Es necesario porque existe la necesidad de asignar o reanudar transacciones entre todas las aplicaciones a las que se está accediendo, orientada por la aplicación de llamadas, que es fundamental. Sin embargo, es complejo porque la restauración para varios sistemas puede requerir la provisión de entradas invertidas o algún otro método basado en el código, que representa un ejemplo ideal de la necesidad de centralizar esta lógica compleja.
Gestión de datos de referencia	Los datos de referencia son comúnmente utilizados por las interfaces de usuarios de una aplicación para proporcionar listas de opciones o para la validación de datos. Es frecuente que los datos de referencia sean comunes entre las aplicaciones (como por ejemplo listas de países, códigos postales, productos, direcciones de oficinas y demás). Acceder datos de referencia desde una ubicación compartida en la capa de integración proporciona un mejor rendimiento, resultados sistemáticos y menos esfuerzo de desarrollo.
Gestión de sesión	La gestión de sesión se utiliza para asegurar que todos los sistemas que comprenden el contexto de integración posean una comprensión activa de quién accede a los datos y para asegurar que las acciones en curso sean sincronizadas entre los sistemas. Se puede realizar un uso completo de la gestión de sesión para permitir que una sesión sea capturada desde un punto de acceso, como por ejemplo un cliente que accede a un sitio en Internet y que sea reutilizada en otro punto como por ejemplo para permitir que un asesor corrija y complete el orden de forma interna.
Instrumentación	De forma similar a la gestión de datos de referencia, la centralización de la instrumentación (registro) dentro de una capa de integración es lógica ya que representa el punto desde el cual se realizan la mayoría de las llamadas con otros sistemas, así como también, permite nuevamente que exista un código de instrumentación en una única ubicación.
Gestión errores	La gestión de errores es similar a la gestión de datos de referencia ya que el uso de una capa de integración para la gestión de errores permite que el código de procesamiento del error sea compartido y permite que se utilice un único conjunto de respuestas de errores.
Gestión de configuración	La gestión de configuración del tipo de entorno descrito aquí es un asunto complejo y digno de su propio contenido. Sin embargo, la gestión de configuración debe proporcionar un punto central en el que las direcciones de los sistemas puedan ser modificadas, el acceso a los sistemas pueda ser controlado (preferentemente con funcionalidad desestimada con elegancia) y en el que la gestión de configuración compartida pueda cargar y almacenar todas las opciones de configuración.

en cuenta las reglas comerciales básicas para permitir que las llamadas únicas combinen aplicaciones.

- **Integración del proceso** – La tercera capa se construye sobre la base de la integración de datos al agregar e integrar los procesos y datos necesarios para ejecutar un proceso de negocio que opera dentro de los límites de la aplicación.

En la medida que uno se desplaza por las tres capas de integración, el objetivo cambia desde la tecnología al negocio. Sin embargo, el resultado final permite una mayor capacidad para agregar valor al negocio de forma inmediata.

Varias vistas de SOA proporcionan un modelo de conectividad común y asumen proporcionar transformación, pero estas vistas sólo proporcionan la integración de datos. Los desafíos que enfrenta un arquitecto al integrar aplicaciones de forma eficaz y

repetible son mucho más complejos, en particular cuando un contexto de integración incluye aplicaciones que ya existen.

¿Por qué todo lo anteriormente citado es importante? Cuando uno construye la primera interfaz entre dos aplicaciones, la simple integración de datos puede ser suficiente. Sin embargo, en la medida que se construyen más interfaces el diseño se vuelve más importante. Si no se diseña y gestiona un contexto de integración completo puede llevar a su vez a un aumento vertiginoso de costos de cada interfaz. Por el contrario, en la medida que crece la funcionalidad que se proporciona en la capa de integración, el código necesario en cada capa puede reducirse.

Componentes de la capa de integración

Los componentes de una capa de integración pueden a su vez dividirse en tres capas: datos, información y proceso (Ver Tabla 1). Además, pueden separarse en componentes que constituyen

Tabla 4: Requisitos de la empresa para la capa de información

Requisito	Descripción
Modelo de información	<p>En la medida en que la cantidad de aplicaciones que están vinculadas dentro de una capa de integración aumenta, también aumenta la cantidad de datos disponibles. Esta situación brinda la posibilidad de comenzar a construir un modelo de información entre los sistemas que se encuentran en el ámbito, que puede ser lo más cercano a crear un modelo de información para una empresa de lo que la mayoría podría lograr.</p> <p>Cuando se implementa de forma correcta, el modelo de información que se coloca dentro de la capa de integración puede proporcionar un punto único de verdad para entidades de datos claves. Un ejemplo de una fuente única de verdad es la comprensión de un cliente que informa todos los sistemas con direcciones, productos, historial de pedidos y mantenimiento.</p> <p>Dos métodos frecuentes para crear un punto único de verdad son: utilizar un sistema clave e intentar sincronizarlo entre otros sistemas o utilizar una base de datos que se actualice con cambios para los sistemas múltiples.</p> <p>El uso de un modelo de información compartida en la capa de integración es particularmente eficaz ya que da como resultado un punto de verdad único compartido que está disponible a través de la combinación de los datos desde fuentes múltiples sin repercutir los datos o códigos que se encuentran en las fuentes. También y debido a esto, no es necesaria la sincronización de datos ya que es más factible que los datos sean actuales.</p>

Figura 5: Requisitos de la empresa para la capa del proceso

Requisito	Descripción
Flujo de trabajo	Los sistemas de flujo de trabajo pueden implementarse como impulsados por el usuario (personas) o impulsados por un sistema. Ambos pueden utilizarse en una capa de integración; aunque, es importante separarlos. El flujo de trabajo impulsado por el sistema incluye la ejecución de los pasos implicados al interactuar con los diversos sistemas en el entorno.
Reglas comerciales complejas	Las reglas comerciales complejas proporcionan un valor adicional en la medida que la capa de integración se completa y en la medida que esto se convierte en el lugar donde las reglas comerciales comienzan a radicarse en su propio derecho. Dada la función central que una capa de integración proporciona, puede convertirse en el host para las reglas comerciales que proporciona funcionalidad extendida o nueva más allá de la integración de aplicaciones, que podrían ser productos nuevos, combinación de productos, descuentos adicionales, una mejor verificación de crédito y demás.
Modelado de procesos de negocio	Los sistemas de flujo de trabajo también pueden utilizarse para la ejecución de flujos de trabajo orientados al usuario, y este motivo es predominante para su existencia. La provisión de una capa de integración brinda el host ideal para que exista el entorno de modelado del proceso del negocio, permitiendo que los flujos de trabajo humanos sean automatizados. El uso de herramientas de flujo de trabajo posibilita que los procesos sean documentados y ejecutados de forma gráfica. El uso de la capa de integración entre las herramientas de flujo de trabajo permite el acceso a aplicaciones múltiples a través de una herramienta de flujo de trabajo más fácil, en particular cuando las capacidades descriptas aquí se han implementado, como por ejemplo el modelado de información y el soporte transaccional. Por lo tanto, se permite el modelado del proceso de negocios para facilitar mejores avances de los que serían posibles.
Monitoreo de la actividad de la empresa (BAM)	El monitoreo de la actividad de la empresa (BAM) comprende el seguimiento de cada transacción por medio de sistemas integrales para identificar obstáculos, rendimiento lento, problemas con transacciones en particular y oportunidades para un desempeño total. En base a estas capacidades de instrumentación que proporciona la capa de integración, las capacidades que brinda un BAM integral son considerables.

un marco técnico; los componentes aparecen en azul en la Figura 2. El *marco técnico* proporciona los requisitos de plomería e ingeniería. Un *marco empresarial* brinda facilitadores de negocios, que se muestran en dorado en la Figura 2. Analicemos más de cerca los componentes que conforman los tres niveles de una capa de integración.

Capa de datos. Los dos componentes de la capa de datos básica son la conectividad y la transformación, que forman los cimientos del marco técnico (Ver Tabla 2). Las interfaces punto a punto únicas que utilizan servicios de Web, con frecuencia se utilizan como ejemplos de SOA, pero varios de los principios de SOA se refieren a la provisión de servicios que empaquetan llamadas múltiples entre los sistemas. Sin embargo, tal como lo mencionáramos con anterioridad, los servicios Web existen en la capa de datos, lo que no facilita el uso de estos principios. En este contexto, los servicios Web permiten la conectividad, pero es necesario explorar la capa de integración para la integración relacionada con el servicio.

Capa de información. La capa de información está compuesta por componentes que se construyen sobre el marco técnico (Ver Tabla 3). El marco del negocio se crea sobre la base del modelo de información (Ver Tabla 4).

Capa del proceso. En la medida que el foco se mueve desde los requisitos técnicos hacia el valor del negocio, el foco de funcionalidad que debería ser brindado a través de la capa del proceso cambia para proporcionar el valor verdadero del concepto de la capa de integración –proceso de negocios configurable de forma rápida (Ver Tabla 5).

Al juntar todo esto, existen varias formas en las que se puede implementar una capa de integración: desarrollo personalizado; uso de programas, paquetes de herramientas, fuente abierta o componentes proporcionados por el sistema operativo y el uso de

“SOA ES COMÚNMENTE PRESAGIADA EN LA ACTUALIDAD COMO LA SOLUCIÓN PARA LA INTEGRACIÓN DE PROBLEMAS ENTRE APLICACIONES, PERO SE REQUIEREN UNA CANTIDAD DE CARACTERÍSTICAS ADICIONALES PARA UNA SOLUCIÓN DE INTEGRACIÓN VERDADERAMENTE EFICAZ”.

un paquete de integración más completo. El desarrollo personalizado se puede aplicar completamente a la capa de integración, pero no se recomienda dada la disponibilidad de paquetes de herramientas y programas.

Algunos programas disponibles brindan la oportunidad de integrar los sistemas de forma inmediata en la capa de interfaz del usuario, como por ejemplo el Customer Care Framework de Microsoft. Este tipo de solución de integración proporciona una solución excelente para la integración inmediata de los sistemas de bases de datos, para que puedan brindar una interfaz del usuario consolidada, pero es importante comprender los méritos relativos de este tipo de solución. Las estructuras que posibilitan la integración de sistemas en la capa de interfaz del usuario proporcionan soluciones rápidas para ciertos problemas que enfrentan las empresas, como por ejemplo mejorar el tiempo de

llamadas para los operadores de un centro de llamadas. Una capa de integración puede proveer muchos más beneficios, a un costo de implementación inicial adicional.

Extender el contexto de integración

Una capa de integración permite que funciones claves se puedan alojar de forma central y que puedan ser compartidas entre las aplicaciones, permitiendo mucho más que la integración de las interfaces del usuario. A su vez, la variedad de datos, información y características de integración del proceso pueden utilizarse entre las aplicaciones para disminuir el costo de cada desarrollo o integración. La experiencia muestra que el uso de una capa de integración tal como se la describe aquí, produce con el tiempo un crecimiento de los ahorros de costos entre una variedad de aplicaciones, con la posibilidad de crear nuevas aplicaciones más allá del contexto de integración original.

Por ejemplo, al facturar, las notificaciones de anomalías, la gestión de clientes y los sistemas ERP pueden integrarse en la primera instancia detrás de una interfaz de usuario CRM, el hecho de que estos sistemas ya estén vinculados proporciona una solución simple para la autoayuda en la Web, y luego se extiende para un pedido online y un canal B2B. ¿Debería existir aquí la adquisición de una nueva aplicación como por ejemplo una aplicación de una cadena de distribución?, la integración de esta

"LAS ESTRUCTURAS QUE POSIBILITAN LA INTEGRACIÓN DE SISTEMAS EN LA CAPA DE INTERFAZ DEL USUARIO PROPORCIONAN SOLUCIONES RÁPIDAS PARA CIERTOS PROBLEMAS QUE ENFRENTAN LAS EMPRESAS".

aplicación dentro del entorno no representa aún un proceso trivial. Sin embargo la integración es mucho más fácil de lo que sería si no se utilizase una capa de integración ya que la integración de los datos puede lograrse de forma más fácil al asociar un único conjunto de interfaces conocidas dentro de la capa de aplicación y no dentro de cada aplicación. También el proceso de integración puede agregarse al extender los flujos de trabajo de integración en el motor de flujo de trabajo, simplemente como integración BAM, agregado de datos, inclusión en un modelo de integración y demás, ya que son mucho más fácil de lograr simplemente extendiendo lo que ya existe.

El componente de flujo de trabajo es una parte clave de cualquier solución de integración que se utiliza para proporcionar tanto el procesamiento de flujo de trabajo centrado en las personas a través de la capacidad de gestión de procesos de negocio, como para desempeñar una función central al ejecutar los procesos comprendidos en los sistemas vinculados. Estas dos tareas están bastante separadas y son bastante diferentes, pero ambas se adaptan perfectamente para el uso de herramientas de flujo de trabajo. Un flujo de trabajo de un sistema de integración puede incluir pasos como recuperar datos desde cada sistema y agregarlos juntos, validar entradas de usuarios y actualizar sistemas conforme a los datos ingresados o actualizados.

Las herramientas de flujo de trabajo proporcionan el método ideal para llevar a cabo estas actividades, son mucho mejores que el uso de códigos y por lo tanto, pueden proporcionar beneficios importantes para la empresa desde una capa de integración, dados los siguientes puntos:

- El trabajo con sistemas múltiples está intrínsecamente orientado a procesos, ejecutando las tareas paso a paso con puntos de decisión, diversificaciones y otros pasos esenciales de flujo de trabajo.
- Los flujos de trabajo de integración de sistemas se sitúan

centralmente entre las aplicaciones y por lo tanto son afectados con más frecuencia por los cambios en todo el entorno, dado que cada cambio para cada sistema reflejará cambios en la capa de integración. El uso de herramientas de flujo de trabajo permite que los procesos sean más fáciles de comprender por muchas más personas que simplemente los desarrolladores originales, dada la naturaleza gráfica y en algún sentido auto documentable de los flujos de trabajo; también permite que los procesos puedan cambiarse de manera más rápida y fácil y que se puedan depurar con más facilidad que los códigos; y posibilita el uso de menos códigos, lo que da como resultado una depuración más fácil y una mayor confiabilidad.

- Dada la forma gráfica que proporcionan las herramientas de flujo de trabajo al brindar una comprensión más fácil de los procesos de la empresa, la propiedad puede ampliarse fácilmente más allá de los equipos de desarrollo, para los analistas del negocio y aún para usuarios centrales que conforman la comunidad de la empresa.

Actitud de Integración Exitosa

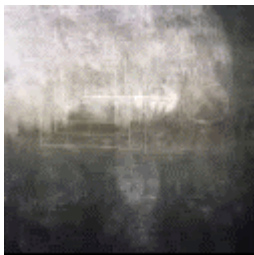
Contar con una herramienta y utilizarla de forma correcta son dos cosas muy diferentes, y este análisis describe una *actitud* necesaria para lograr una integración realmente exitosa. Independientemente del método que se utilice, recuerden que si se tiene una visión de la integración en toda la empresa se podrán lograr beneficios mucho más grandes para los negocios, la integración en sí misma no es trivial y debe considerarse dentro de un contexto de empresa; y mientras se considera el contexto de la empresa y se diseña para obtener los mayores beneficios del negocio es aún posible –y necesario– comenzar a implementar la solución en un proyecto único.

La visión empresarial de la integración debe considerarse lo antes posible. Si bien existe claramente un lugar para la integración punto a punto en el que no es posible una integración adicional, un contexto de integración más capaz que proporcione el flujo de trabajo y otras características descriptas aquí, brindará ventajas en la medida que crezca la cantidad de sistemas e integración. Por lo tanto, es necesario un análisis fuera del ámbito de cada proyecto único para descubrir cualquier oportunidad que pueda existir.

Del mismo modo que con todas las decisiones de arquitectura, existen características, costos, orientadores del negocio y demás que se tienen en cuenta al tomar decisiones de arquitectura. El enfoque para la arquitectura de integración que se recomienda aquí es muy conveniente para las características y conceptos de arquitectura de la actualidad. Esperamos que le ayuden a comprender algunas de las decisiones que pueden tomarse en el área. ●

Sobre el autor

Kevin Francis es profesional IT con 19 años de experiencia en la industria en una variedad de puestos –desde CIO en una empresa de fabricación, analista de seguridad, hasta arquitecto global– y también dirigió su propia y exitosa empresa de desarrollo de software. Kevin ha diseñado y administrado proyectos de comercio electrónico de vanguardia desde que surgió el comercio electrónico. Como arquitecto principal de Infosys, Kevin es responsable del éxito de la implementación técnica de la compañía para sus clientes. Asesora a grandes empresas, trabaja con los arquitectos de Infosys, equipos de desarrollo y clientes para asegurar que cada solución técnica incorpore calidad y dirección estratégica del cliente. Kevin es miembro de *The Infosys Australia Technology Council* y fue galardonado por Microsoft con el premio *Most Valuable Professional* (MVP) en julio de 2005 por su conocimiento y experiencia como Desarrollador Visual y Arquitecto de Soluciones.



Simplificar el diseño de flujos de trabajo complejos

Por Andrew Neddleman

Síntesis

Diversos factores tratan el diseño de un flujo de trabajo adecuado. Para simplificar el proceso, es necesario concentrarse sobre la diagramación de los pasos necesarios para una transacción clave en particular como por ejemplo una orden de comercio electrónico o una consulta médica. En este debate analizaremos la representación del flujo de trabajo en un tipo de diagrama nuevo, denominado diagrama del "proceso de puntos y líneas". Comenzaremos diagramando los posibles pasos que pueden tomarse en cada punto del flujo de trabajo y luego indicaremos el punto donde nos lleva el diagrama en este proceso de flujo de trabajo. También nos aseguraremos de no olvidarnos en el camino ninguno de los puntos de estado o pasos potenciales.

Un buen diseño de flujo de trabajo requiere el análisis del proceso de negocios, el rediseño del proceso de negocios, el análisis de utilidad y el diseño del software. Esta lista desalentadora de habilidades necesarias hace que el flujo de trabajo sea un desafío para la gran mayoría de los arquitectos de software expertos. ¿Cómo desglosar el problema en las más pequeñas partes para comenzar en la dirección correcta? Analizaremos la forma de simplificar el proceso de diseño de sistemas complejos con tipos de usuarios múltiples y cientos de estados potenciales. Para simplificar el proceso se puede pensar en una solución de flujo de trabajo más que en un proceso de diseño de flujo de trabajo.

Cada proyecto de software posee un proceso central que es la razón principal por la cual existe el sistema o por la cual se está creando. Por ejemplo, el objetivo principal de un sitio de comercio electrónico es facilitar a las personas la compra de productos. El proceso principal es el que necesita la mayor atención durante su diseño e implementación. Les mostraremos la *representación con puntos y líneas* para modelar lo que lo ayudará a comunicarse con los expertos de negocios sobre procesos complejos para determinar los estados y acciones disponibles en la medida que crea el flujo de trabajo.

Comencemos con una visión general de la representación con puntos y líneas para la creación de diagramas de flujo de trabajo. En la representación con puntos y líneas cada *punto* representa un estado particular en el proceso, y cada *línea* corresponde a una acción que puede ser tomada desde este punto. Los puntos y las líneas están etiquetados y se genera una clave en la medida que se desarrolla el flujo de trabajo. Los puntos están etiquetados con letras, comenzando por la A, y las líneas están etiquetadas con números, comenzado por el 1. De esta forma, las líneas y los puntos no se mezclan.

Los objetivos primarios detrás de la representación con puntos y líneas son crear una herramienta de visualización simple sobre papel que pueda utilizarse con los usuarios de la empresa, eliminar el perfeccionismo que causa la diagramación por computadora, hacer que el proceso sea lo más liviano posible como para permitir la concentración sobre la difícil tarea de diseñar el flujo de trabajo y separar los cambios en la descripción de acciones y estados de los cambios en el flujo del proceso (es decir, similar a la idea de una tabla de búsqueda).

Seguir las reglas

Además de estos objetivos, existen cinco reglas que deben aplicarse un diagrama que utiliza esta representación. Analicemos los detalles de cada regla.

El diagrama siempre va en una dirección. Una de las ideas principales de la representación con puntos y líneas es que el diagrama siempre se lee hacia adelante como una escala de tiempo. Se puede finalizar en un estado previo pero el diagrama no posee líneas que van en las cuatro direcciones como un laberinto.

Los estados pueden descubrirse en la medida que se crea el diagrama. En un flujo de trabajo simple se conocen todos los estados potenciales antes de comenzar la diagramación. Sin embargo, en un flujo de trabajo más complejo es necesario descubrir los estados potenciales al construir el diagrama. Determina las acciones que puede tomar en cada punto del proceso y ve dónde lo lleva. En la medida que se descubren los estados se puede decidir la forma de etiquetar cada estado para su uso. Entonces, una vez que se ha completado el diagrama, se puede decidir la manera de presentar los estados a cada tipo de usuario y si es necesario combinarlos o no en estados que se comprendan con más facilidad.

"CADA PROYECTO DE SOFTWARE POSEE UN PROCESO CENTRAL QUE ES LA RAZÓN PRINCIPAL POR LA CUAL EXISTE EL SISTEMA O POR LA CUAL SE ESTÁ CREANDO."

Las etiquetas de los estados y acciones no aparecen en el diagrama. Si se tienen descripciones de cada acción y estado, el diagrama no es simple y es difícil de comprender. Para colocar etiquetas para los estados y acciones en una clave, es mucho más fácil ver flujos de procesos más largos en espacios más pequeños, lo que sería casi como normalizar el diagrama. Se pueden cambiar las etiquetas de los puntos sin tener que cambiar el diagrama. Esto posibilita una más fácil visión y diagramación de procesos complejos.

La diagramación es simple y flexible, permitiendo que todos puedan precisar con claridad el contenido. La diagramación es un gran problema en los flujos de los procesos tradicionales. Se debe

realizar mucho esfuerzo para conectar y ajustar los diferentes ítems dentro del diagrama. Si se omite una acción potencial en un flujo del proceso, entonces tal vez sea necesario reorganizar el diagrama completo. En la representación con puntos y líneas sólo es necesario encajar una línea para una acción y un punto para el estado del resultado. Si es necesario cambiar una acción o estado, se puede cambiar el número o la letra respectivamente. Si se desea cambiar el nombre de un estado o acción, simplemente se debe cambiar la clave en vez del diagrama. La simplicidad de agregar, cambiar y visualizar puntos y líneas en el diagrama hace más fácil mantener la atención en el proceso en sí mismo.

“LOS ESTADOS Y LAS ACCIONES DEBEN INCLUIRSE SI SON NECESARIOS PARA COMPLETAR EL PROCESO QUE SE ESTÁ ESTUDIANDO”.

Los estados se pueden “separar” del diagrama principal. Debido a que siempre se va hacia adelante en el diagrama, es posible separar secciones del diagrama sin preocuparse sobre la forma que se conectarán a estados previos en el flujo de trabajo. Por ejemplo, A se conecta a B, y queremos separar B de su propio diagrama porque posee demasiadas descendencias para ajustar en el diagrama existente. No es necesario preocuparse si B posee una acción que va a C y C posee una acción que envía su estado de regreso hacia A. Al etiquetar el estado con A están conectadas, en vez de tener que reenviar una línea al estado A original. El diagrama sólo se mueve hacia adelante, no hacia atrás, por lo tanto, no es necesario volver a conectar con el punto A original. Esta regla permite una gran flexibilidad cuando uno se está quedando sin espacio en un diagrama.

Acciones y estados para flujos de trabajo

Ahora trataremos algunas de las ventajas de la representación con puntos y líneas, veamos la forma de determinar si una acción o estado debe estar en el diagrama. Los estados y las acciones deben incluirse si son necesarios para completar el proceso que se está estudiando. Por ejemplo, si bien iniciar una sesión o registrarse en un sistema de comercio electrónico no cambia el estado del objeto de una orden, esto es necesario para completar la compra. Por lo tanto, debe incluirse como una acción en el sistema y los estados que crean estas acciones deben estudiarse y diferenciarse si se está analizando el proceso de órdenes.

Los estados y las acciones no deben incluirse si no son necesarios para completar el proceso que se está estudiando y no cambian los pasos necesarios para completar el proceso que se está modelando. En otras palabras, una acción que lo hace retroceder un paso en el proceso es algo que debe modelarse. Por otro lado, una acción que no está relacionada al proceso no debe modelarse.

Por supuesto, existe definitivamente un gran margen de interpretación sobre qué se debe incluir en el diagrama del proceso; debido a que es necesario encender la computadora para realizar cualquier proceso informático, es probablemente un poco ridículo incluir esta acción como parte de un diagrama de proceso. Ahora utilizaremos un ejemplo para tratar de clarificar lo que se debe y lo que no se debe incluir en un diagrama con puntos y líneas.

Analicemos un ejemplo conocido de un flujo de trabajo para examinar este proceso –la compra de productos a través de un sitio Web de comercio electrónico. Estudiaremos los pasos necesarios que se deben seguir para comprar un producto.

Cuando comenzamos a diseñar un flujo de trabajo de compras, ignoramos pasos que no cambian el estado de la transacción de compra. Por ejemplo, el proceso de búsqueda de un producto no cambia en absoluto el estado del proceso de la orden. Realmente

Tabla 1: Clave para el diagrama de una orden de comercio electrónico (Ver Figura 1)

Acciones	
1	Encontrar y agregar un producto en el carro de compras.
2	Registrarse.
3	Iniciar la sesión.
4	Eliminar todos los productos del carro de compras.
5	Cerrar la sesión.
6	Iniciar el proceso de pago.
7	Agregar/seleccionar información sobre el envío.
8	Agregar/seleccionar información sobre la facturación.
9	Procesamiento de la facturación exitoso.
10	Procesamiento de la facturación falló.
11	Confirmar orden.
12	Salir del proceso de pago.
Descripciones del estado	
A	No hay productos en el carro de compras, no se ha iniciado la sesión.
B	Los productos están en el carro de compras, no se ha iniciado la sesión.
C	No hay productos en el carro de compras, sesión iniciada.
D	Los productos están en el carro de compras, sesión iniciada.
E	Iniciar el proceso de pago, no se ha iniciado la sesión, sin envío ni facturación.
F	Iniciar el proceso de pago, sesión iniciada, sin envío ni facturación.
G	Proceso de pago, sesión iniciada, envío, sin facturación.
H	Proceso de pago, sesión iniciada, envío y facturación.
I	Pago confirmado, sesión iniciada, estado temp (cuadrado).
J	Procesamiento de la orden exitosa, sesión iniciada.

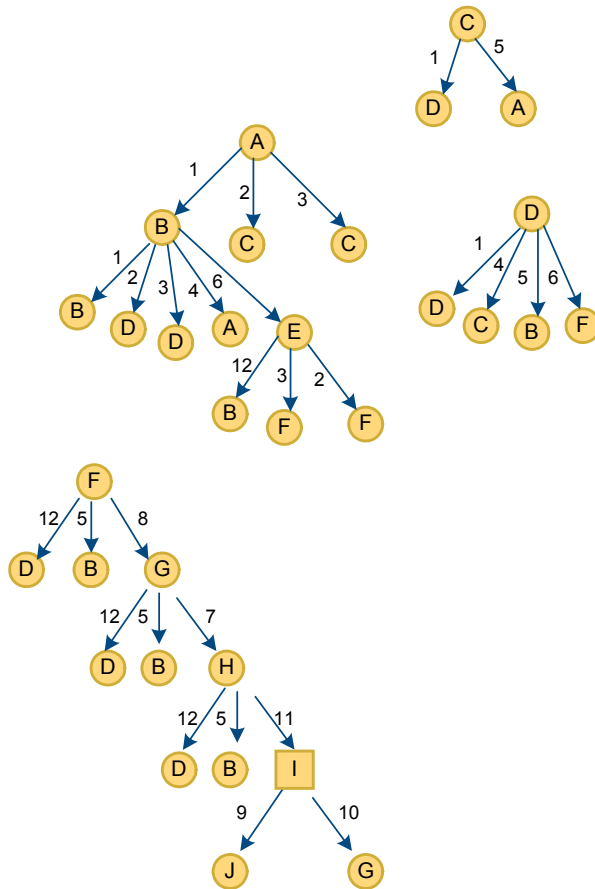
no importa la forma en la que el usuario se mueve por el sitio a menos que realice una de las tres cosas que cambian su estado en curso. Estos tres pasos que pueden hacer avanzar el estado de compra del producto son: agregar un producto en el carro de compras, registrarse en el sitio e iniciar la sesión en el sitio.

Por ejemplo, supongamos que vamos a la sección “libros” y hacemos un click hacia atrás para regresar a la página principal. ¿Cambia esta acción de alguna manera nuestro estado en lo que se refiere a la compra del producto? No. Imaginemos que realiza una búsqueda del sitio “.NET books”, y se obtiene como resultado una lista de ellos. Luego se realiza un click hacia atrás para regresar a la página principal. Aún no se ha avanzado en el proceso de compra.

Incluso si suponemos que contamos con un motor que coincide con las sugerencias de libros del historial de búsqueda, aún así no se avanza en el proceso de compra hasta que no se agrega un producto en el carro de compras. Es muy importante que este paso esté fuera del diagrama porque de otra forma lo complicaría innecesariamente. Podemos combinar la búsqueda y luego agregar un producto en el carro sólo en una acción en el diagrama del proceso de compra, denominada “buscar y agregar producto”. Si deseáramos diagramar el proceso de búsqueda de un producto, entonces estaríamos interesados en la manera en la que el usuario

Representación con puntos y líneas para flujos de trabajo simples

Figura 1: Diagrama de representación con puntos y líneas para el proceso de la orden de un producto en un comercio electrónico (Ver la Tabla 1 para referencia de las claves)



selecciona el producto para verlo y agregarlo a su carro de compras. Por ahora, nos concentraremos en diagramar el proceso de compras.

Funciones en el proceso de compras

Primero, comenzaremos con alguien que simplemente ingresó en la página inicial de nuestro sitio de comercio electrónico. Analizaremos este caso para ver qué sucede después en nuestro proceso. Tres posibilidades en cuanto a comportamiento que avanza el proceso de compras son: agregar un producto en el carro de compras, registrar un usuario e iniciar la sesión en el sitio. Se puede ver el diagrama de puntos y líneas para este proceso completo en la Figura 1 y sus claves respectivas en la Tabla 1.

Es importante comprender el impacto que tienen las múltiples funciones en el sitio sobre los estados y permisos visibles. Por ejemplo, supongamos que nuestro sitio de comercio electrónico tiene funciones tanto para los compradores como para un administrador. Los compradores son las personas que compran el producto, mientras que el administrador se asegura de que obtengan el producto. El diagrama de puntos y líneas nos ayuda a

determinar lo que se debe mostrar al administrador proporcionándole una lista integral de los estados en los que se encuentra una orden durante el proceso de compra. Podemos entonces tomar estos estados y ejecutar la lista para averiguar cuáles son las órdenes que debe ver el administrador y cuáles no. Por ejemplo, una de las reglas del negocio potencial puede ser que el administrador pueda ver las transacciones de órdenes que se abandonan en la solicitud de facturación para permitir que él realice un seguimiento de esos clientes.

Para sitios más complejos que poseen cientos de estados potenciales para cada transacción que se realiza en el sitio, es necesario acumular estados múltiples en descripciones de estados únicos. Por ejemplo, los estados E-G pueden agruparse como

“CUANDO COMENZAMOS A DISEÑAR UN FLUJO DE TRABAJO DE COMPRAS, IGNORAMOS PASOS QUE NO CAMBIAN EL ESTADO DE LA TRANSACCIÓN DE COMPRA”.

estado “incompleto” para el administrador. Es poco probable que el administrador necesite saber más que esto, excepto para estadísticas globales que se proveerían en un informe más que en una pantalla de trabajo diario. Además, si el administrador entra en alguna de estas órdenes puede proporcionar detalles del estado en ese punto.

Una de las formas más fáciles de acumular estados para cada usuario es utilizar el método “¿quién tiene la pelota?”. En otras palabras, ¿Qué usuario se supone que debe hacer algo para avanzar en el flujo de trabajo? El/Los usuario(s) que se adecuen a esta descripción deben ver este ítem agrupado con otros ítem que están esperando alguna acción de su parte. Luego, se pueden agrupar otros ítems abiertos en otra categoría y finalmente los ítems cancelados/finalizados en una tercera categoría.

Una recomendación final con respecto a la representación con puntos y líneas es utilizar un lápiz y un papel bien grande para crear los diagramas. Otro papel bien grande debe contener los nombres de los estados, y por último, se deben tener los nombres de las acciones. Utilizar hojas de papel separadas y bien grandes de esta manera permite que todos tengan un buen panorama general del proceso completo en vez de tenerlo por separado en múltiples pantallas.

El uso de papel hace que todos se enfoquen en el problema en cuestión en vez de que parezca que todo está normal. Si se utiliza papel no existe forma fácil de poner se puede poner con facilidad un gradiente sobre cada punto o de reorganizar los ítems para que encajen mejor. Usted debería encontrar esta representación con puntos y líneas tan útil como yo, al descubrir todos los estados y acciones potenciales con los expertos de negocios. •

Sobre el Autor

Andrew Needleman es socio gerente en Claricode (www.claricode.com), desarrollador de soluciones personalizadas exclusivamente para la industria del cuidado de la salud. Ha escrito artículos que se centran en el desarrollo del software para varias publicaciones IT e industrias del cuidado de la salud. Su experiencia en el diseño de flujos de trabajo incluye el diseño de una aplicación con cientos de estados diferentes, acciones múltiples en cada estado y nueve tipos de usuarios distintos. Esta solución fue reconocida como una *Intel Solution Blueprint* para las mejores prácticas en el cuidado de la salud con tecnologías de Microsoft. Póngase en contacto con Andrew en andrew@claricode.com.

Habilitación de la Empresa orientada al servicio

Por William Oellermann

Síntesis

La construcción de servicios Web es bastante fácil actualmente. La construcción de numerosos servicios Web es un poco más difícil, y administrarlos es realmente difícil. Al aumentar la cantidad de servicios y consumidores del servicio, los beneficios fundamentales de la orientación al servicio disminuyen si no se brindan ciertas capacidades de la informática. Echemos un vistazo a un modelo que puede facilitar la identificación y priorización de estas capacidades para empresas habilitadas para la prestación de servicios.

Cuando se habla con clientes y colegas sobre los servicios Web, se descubre que mucho de la excitación e interés se enfocan en las especificaciones que han ayudado a esta tecnología a transformarse rápidamente en una norma de facto: SOAP, WSDL, XML, WS-Security, y demás. Aunque esta estandarización es fundamental, con frecuencia nos distrae del siguiente nivel de discusión sobre la manera de aprovechar real y completamente este paquete común. Estos protocolos nos dan un mayor alcance y reutilización que el que alguna vez hayamos logrado con el software, pero su uso individual no cumple ese potencial.

Todo el trabajo invertido en normas y consorcios nos permite desarrollar servicios que puedan interoperar con otras aplicaciones y sistemas, pero no necesariamente nos otorga la reutilización. Lo que es aún más importante, no nos otorga la manejabilidad. Si se trabaja en una organización, o con una organización, que obtuvo algún éxito con servicios Web, no habrá dudas de que se encontrará la necesidad de tratar, (cuando no la desventaja de no hacerlo), las pruebas de servicios, implementación, versión y administración. Estas áreas son muy problemáticas cuando no se identifican antes de construir una cartera de servicios.

Esta condición ha convertido ya a algunas organizaciones en víctimas de su propio éxito. Los enfoques que trabajan con muchos servicios con frecuencia fallan al aumentar la cantidad de servicios y su uso. Si no se tienen los principios y prácticas en lugar de promocionar la reutilización y brindar manejabilidad al aumentar el uso de servicios, se encontrará el "techo de escalabilidad". (Ver la barra lateral, "Aclaración sobre Modelos de Maduración").

Debatir este tema es muy difícil, incluso entre los arquitectos más experimentados e inteligentes, mucho menos otros participantes menos técnicos que juegan un rol en el ecosistema de servicios. Algunas de estas dificultades provienen del uso de términos sobrecargados como servicio, pero mucho de esto proviene de la complejidad pura que está implicada con una arquitectura distribuida en forma masiva que toca muchas áreas de una organización. Si algo es difícil de debatir, sólo se debe pensar en cuán difícil puede ser definir una estrategia o planearla.

Estos desafíos nos llevan a ver el uso de un modelo para ayudar a facilitar la discusión de habilitación para la prestación de servicios y planificación de una empresa habilitada para prestación de servicios. Con los aportes y feedback de muchas organizaciones y colaboradores atentos durante los últimos dos años, hemos desarrollado el Modelo de Maduración de Orientación al Servicio en Empresas (ESOMM).

Un modelo de maduración impulsado por la capacidad

Las 4 capas, 3 perspectivas y 27 capacidades definidas en el ESOMM están diseñadas como un mapa de ruta para soporte de servicios, no un servicio específico con un uso, implementación o aplicaciones específicos, sino cualquier servicio, o en particular, cualquier conjunto de servicios. Si se intenta desarrollar sólo un servicio, el valor que este modelo brinda será limitado. El objetivo de ESOMM es ayudar a solucionar el problema de la escalabilidad en un nivel de grupo, departamento, división o empresa. Este objetivo significa que se puede aplicar como parte de una iniciativa de base o una estrategia empresarial. La habilidad que se tenga para brindar ciertas capacidades puede verse limitada por la cantidad de alineación organizativa total, pero ese factor no debería impedir la capacidad de comenzar a construir y a implementar un plan que impulse ESOMM.

Desarrollar una estrategia de habilitación para prestación de servicios no es un compromiso trivial y no se debe considerar como un esfuerzo a corto plazo o de una sola vez. En su cumbre, la orientación al servicio tiene el objetivo de permitir un más alto nivel de agilidad para una organización, para que pueda proveer una respuesta rápida a las necesidades de los negocios y sus

"TODO EL TRABAJO INVERTIDO EN NORMAS Y CONSORCIOS NOS PERMITE DESARROLLAR SERVICIOS QUE PUEDEN INTEROPERAR CON OTRAS APLICACIONES Y SISTEMAS. PERO ESTO NO NECESARIAMENTE NOS OTORGA LA REUTILIZACIÓN".

clientes. El logro de esta agilidad significa brindar un plan que sea extensible y duradero al mismo tiempo. Al enfocarse en el servicio, y no en el mensaje, implementación, o uso, ESOMM puede ayudar a construir un plan que cualquier organización pueda aplicar ampliamente. Como implementaciones de servicio preferenciales, elecciones de tecnología, y modificación de patrones de uso, el modelo deberá proveer un mapa de ruta duradero que pueda soportar esos cambios.

ESOMM consiste en cuatro capas, cada una con un conjunto de capacidades que como un todo establece un nivel sólido de logro y valor para la organización para esa capa (ver Figura 1). Este valor debe realizarse completamente si se proveen todas las capacidades adjuntas. Es posible, y ciertamente probable, que una organización pueda tratar partes de capas múltiples;

el objetivo no es satisfacer una capa, sino identificar el nivel de valor que desea una organización.

La primera capa de maduración en ESOMM, llamada *utilizable*, consiste en el uso adecuado de normas y protocolos para diseño y desarrollo de servicios que son utilizables en todas las plataformas y tecnologías de la organización. Construir los servicios en forma *correcta* es absolutamente fundamental antes de que se puedan establecer exitosamente muchas de las capacidades de alto orden en el conjunto SOA. La verdadera prueba de cuán bien se construyen los servicios es el grado de consistencia presente en todos los servicios, independientemente de las herramientas y tecnologías utilizadas.

La capa *repetible* trata muchas de las capacidades que hacen a los servicios eficaces de desarrollar, implementar y mantener en la organización. Sin tratar capacidades clave en esta área, será muy difícil escalar exitosamente el uso de servicios Web.

La capa *soportable* engloba muchas de las capacidades necesarias para brindar en forma confiable servicios para aplicaciones de misiones críticas que puedan soportarse. Estas capacidades incluyen el soporte operativo de servicios, pero lo más importante, capacidades de auto servicio brindadas a los consumidores. El autoservicio es una herramienta importante que permite que los servicios escalen evitando que los desarrolladores y arquitectos pierdan tiempo en correspondencia con consumidores potenciales de los servicios. Esta capa es muy difícil de tratar en forma satisfactoria sin tener primero éxito al hacer a los servicios repetibles en su implementación y administración.

La capa *extensible* representa el pináculo de la realización de la agilidad comercial prometida por los servicios y la reutilización de servicios como bloques que construyen otros servicios o agregado de estos. Esta noción en general implica brindar capacidades a los clientes y/o socios mediante servicios que generan nuevos canales de ganancias, si no nuevos modelos comerciales. Existen varias capacidades requeridas para que esta realización sea posible, y compensan la mayoría de los focos en esta capa extensible.

Perspectivas del mapa de ruta

Mediante compromisos e interacciones con clientes en los últimos años, reconocimos una necesidad de tratar diferentes perspectivas mediante este modelo. Al hablar de las capacidades en cada capa del modelo, se volvió evidente que había variables niveles de apreciación o consideración para ellos, según el rol y desafíos del público. Por ejemplo, una organización tenía dificultades en apreciar la necesidad de diseño de interfaz por contrato. Estaban bastante satisfechos utilizando herramientas para generar dinámicamente contratos de servicio y sentían que tenían éxito con este enfoque.

Cuando hablamos con los usuarios de los servicios del grupo, no obstante, tuvimos una perspectiva totalmente diferente al saber que el uso de tipos de datos específicos y patrones de elementos causaba muchas dificultades durante la implementación. Al reconocer que no todas las capacidades son impulsadas por propietarios o proveedores del servicio, los inversores pueden tener mayor apreciación de algunas de las capacidades y tomar decisiones más informadas sobre el enfoque y priorización de estas capacidades.

Las capacidades en cada capa de ESOMM están clasificadas en una de estas tres perspectivas: implementación, consumo, y administración (ver Figura 2). Las capacidades de *implementación* apuntan al desarrollo y la implementación de servicios Web desde la perspectiva del proveedor. Esta perspectiva es una que la mayoría de las organizaciones aprecian por defecto, y exige la menor cantidad de clarificación. Las capacidades de *consumo* son aquellas que satisfacen a los consumidores de los servicios, facilitándoles la implementación y por lo tanto son adoptadas más

Figura 1: Las cuatro capas de maduración de ESOMM

Extensible	Una empresa es capaz de incorporar servicios y extender su uso, más allá de sus propios límites.
Soportable	Una empresa puede administrar eficazmente cantidades crecientes de servicios a los SLAs garantizados.
Repetible	Una empresa implementa, consume y reutiliza servicios en forma eficiente y consistente.
Utilizable	Una empresa es capaz de escribir y consumir servicios conforme a normas con un excelente alcance.

ampliamente y con mayor éxito. Creemos que esta es la perspectiva que más se descuida, pero puede ser la más crítica para determinar el éxito con la orientación al servicio. Si es difícil impulsar los servicios, el éxito será limitado, sin importar qué otras áreas estén optimizadas. Las capacidades de *administración* utilizan aspectos operativos y de régimen de servicios Web en la organización. Esta perspectiva se vuelve más reconocida y apreciada, pero todavía hay más para entender sobre su valor e impacto.

El valor adicional de las perspectivas es la ayuda que brindan para priorizar la implementación de capacidades específicas en el modelo. Aunque hay una habilidad de ignorar ciertas capacidades

"LA CAPA EXTENSIBLE REPRESENTA EL PINÁCULO DE LA REALIZACIÓN DE AGILIDAD COMERCIAL PROMETIDA POR LOS SERVICIOS Y LA REUTILIZACIÓN DE SERVICIOS COMO BLOQUES DE CONSTRUCCIÓN DE OTROS SERVICIOS, O INCORPORACIÓN DE SERVICIOS".

al progresar en el modelo, es bastante arriesgado saltar capacidades a una capa menor dentro de la misma perspectiva. Estos pre-requisitos no sólo tienen la intención de ayudar a hacer crecer el sistema de manera lógica, sino también de ayudar a saber cómo utilizar los servicios y soportarlos en la organización, porque esta información impactará el diseño de ciertas capacidades a capas superiores.

Es también importante darse cuenta de que una organización puede elegir enfocarse en una o dos perspectivas, pero la maduración total, y por lo tanto el valor total de los servicios, depende del nivel adecuado de atención a las tres. Descuidar cualquiera de las tres tendrá consecuencias, algunas más obvias que otras. Tales consecuencias serían semejantes a ignorar uno de los grupos alimenticios; se puede hacer, pero limitaría algunas de las comidas que se pueden disfrutar y que no son ideales para la salud general.

Capacidades del ESOMM

Existen 27 capacidades descriptas en el modelo ESOMM, y fácilmente tomaría un artículo explicar cada una en detalle (ver figura 3). Aquí daremos un panorama breve de estas y ayudaremos a articular la premisa de cada capacidad y su conformidad con la perspectiva y capa asociada.

La perspectiva de implementación incluye los procesos de desarrollo y capacidades de patrones de diseño en la capa utilizable (1):

Procesos de desarrollo. Como los patrones de diseño, los procesos de desarrollo se dirigen a un concepto con el cual los grupos de desarrollo se encuentran muy familiarizados, pero se enfoca específicamente en la implementación de servicios. A un nivel básico, este concepto generalmente consistiría en el uso de normas comunes soportadas en las más comunes plataformas y herramientas, incluso SOAP, WSDL, protocolos WSA, y perfiles WS-I. A un nivel más avanzado, la capacidad incluiría la definición de procesos repetibles para el diseño de interfaces de servicio, implementación y publicación de servicios. Nuevamente, el objetivo es hacer que estos procesos estén ampliamente disponibles en todos los límites de la plataforma.

Patrones de diseño. Cuando se desarrolla el software, es bueno aprovechar los patrones de diseño *probados* que son adecuados para esa implementación. Esta capacidad en particular se dirige a los patrones de diseño que son adecuados y necesarios, específicamente para uso con servicios. Existe buena información disponible en este campo, comenzando con los cuatro principios de orientación al servicio (ver Recursos). Otros ejemplos de patrones importantes incluyen ejecución asincrónica, soporte de devolución de llamada, manejo de excepciones, y uso de datos de encabezado SOAP. La clave para cumplir con esta capacidad es definir patrones que los desarrolladores puedan utilizar en todas las plataformas y herramientas adecuadas.

Desde la perspectiva de consumo del nivel utilizable existen contratos explícitos y capacidades para evaluación.

Contratos explícitos. La información más básica que hay que proveer a los consumidores de servicios son los contratos explícitos, que incluyen no sólo diagramas bien detallados (aumentando el poder de XSD), sino datos de política que comparten información de contexto adecuada. Esta información permite que se sepa no sólo la composición del cuerpo de carga explosiva del mensaje, sino los elementos de encabezado que se soportan y son necesarios. La dificultad de esta capacidad es que hay poco soporte para algunos de los estándares emergentes alrededor de la política compartida, así que exige algo de análisis y esfuerzo para tratarlo adecuadamente.

Evaluación. Esta capacidad brinda a los consumidores un mecanismo para evaluar la invocación de un servicio para soportar la implementación y solución de problemas desde su lado. Uno de los aspectos más difíciles de implementar sistemas distribuidos es la prueba de integración. Con tantas dependencias, se puede necesitar mucha coordinación para simplemente finalizar una llamada a través de toda una cadena, incluso para una operación nula o un

pedido inválido. Los servicios Web están en una posición única para abstraer aplicaciones desde sistemas back-end que realizan el procesamiento; un entorno de evaluación, ya sea físico o virtual, puede ser una herramienta estratégica para mejorar la eficacia de implementación y reducir la necesidad de soporte manual.

La perspectiva de administración de la capa utilizable contiene el monitoreo básico y capacidades de modelo de seguridad.

Monitoreo básico. El soporte de servicios Web comienza con el soporte para hosting de infraestructura básica y provisión de esos servicios. Este soporte incluye lo básico de la excelencia

"UNO DE LOS ASPECTOS MÁS DIFÍCILES DE IMPLEMENTAR SISTEMAS DISTRIBUIDOS ES LA PRUEBA DE INTEGRACIÓN".

operativa, que incluye monitoreo de contadores y registros clave que puedan indicar temas de procesamiento y dar alertas para indicar cortes de servicios importantes. Esta capacidad no incluye visibilidad a nivel de servicio o monitoreo, porque no es un compromiso trivial y seguirá ascendiendo el conjunto del modelo.

Modelo de seguridad. Una vez que se adopta un servicio, las organizaciones verán que se usan estos servicios en muchas aplicaciones, tanto externas como internas. Es importante tener un modelo estándar de seguridad para soportar los escenarios principales para cumplir con requisitos comerciales. Este modelo implica crear una o más tiendas de credencial que se puedan administrar y soportar como parte del ecosistema de servicios. Los ejemplos de estas tiendas de credenciales incluyen autoridades de certificación, directorio de LDAP, servicios de Active Directory, y bases de datos. Las credenciales son un componente fundamental del modelo de seguridad de servicio, por lo cual tratar completamente esta capacidad también significa soportar los requisitos en el acceso, autenticación, autorización y encriptado. Estos enfoques no necesariamente deben implementarse como características, si no son necesarios en ese momento, pero se debe definir un modelo de seguridad completo para cuando llegue el momento. Esta implementación será seguramente el primer paso para definir la política de servicio.

Capacidades de la capa repetible

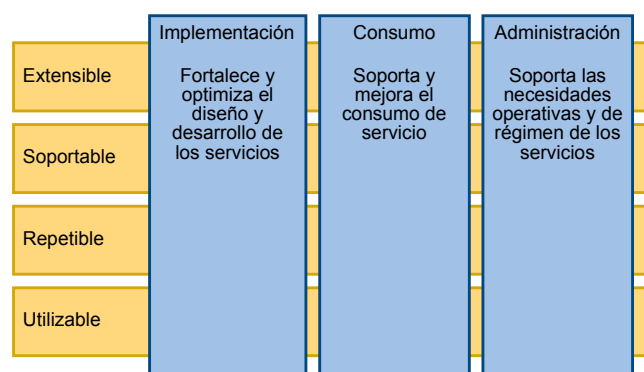
La perspectiva de implementación incluye bloques de servicio y capacidades de esquema comunes en la capa repetible (2):

Bloques de servicio. Una clave para reducir la cantidad de lógica de "rutina" colocada en cada servicio es el uso de un concepto llamado bloques de servicio. Esta capacidad es un modelo de arquitectura que provee rutinas comunes que se deben impulsar ampliamente como transformaciones, búsquedas de punto-datos, y procesos de utilidad. Un ejemplo de bloque de servicios sería un servicio de registro, y de hecho nuestro bloque de aplicación de registro podría ser habilitado para la prestación de servicios a fin de cumplir este requisito. Para que tenga éxito este modelo, estos bloques de servicio deben estar disponibles para todos los servicios de la organización.

Esquema común. Cuando se proveen servicios que funcionan con entidades comunes, es útil tener una definición común para que esas entidades minimicen la cantidad de esfuerzo necesario para brindar y consumir esos servicios. Debido a los sistemas múltiples generalmente involucrados en proveer lógica comercial y datos para servicios, esta definición generalmente es un esfuerzo continuo que no necesariamente será considerado *completo*. No obstante, incluso establecer definiciones comunes para muchas entidades centrales puede obtener tremendos beneficios para los esfuerzos de habilitación para la prestación de servicios.

La perspectiva de consumo para la capa repetible incluye la capacidad de descubrir el servicio y la capacidad de auto suministro:

Figura 2: Tres perspectivas sobre las capas de maduración



Capacidad de descubrir el servicio. Una vez comenzada la provisión de servicios, descubrirlo se convierte en capacidad clave para ayudar a guiar a los consumidores a identificar el servicio adecuado para usar. Esta guía se puede lograr manualmente o con programas con herramientas (por ejemplo, mediante UDDI). Lamentablemente sólo brindar una lista de servicios en una página Web no prueba nada a este nivel, así que una de las claves de guía del proceso de descubrimiento es la clasificación inteligente. Una o más taxonomías bien pensadas pueden ayudar a los consumidores potenciales a buscar intuitivamente los lugares correctos de servicios en base a dominios o implementación específica.

Auto suministro. Al desarrollar más servicios, se volverá más difícil para los consumidores solicitar acceso a estos. Aunque el proveedor puede desear automatizar el back end del proceso de provisión al usuario, es más importante primero permitir a los consumidores identificar y activar rápidamente las herramientas disponibles para solicitar acceso. Si este proceso es demasiado difícil o problemático, el índice de adopción de estos servicios se verá afectado, limitando el éxito integral. Esta capacidad se logra con una amplia gama de soluciones desde la más simple a la más compleja.

La perspectiva de administración del nivel repetible incluye las políticas empresariales y capacidades de administración de implementación.

Políticas empresariales. Las políticas de servicio son importantes para definición, soporte y refuerzo al aumentar el uso de servicios de la organización. El punto de inicio para esto fue desarrollar un modelo de seguridad, pero a este nivel es necesario ampliar la política para soportar un conjunto de opciones posibles disponibles según los diferentes criterios de un escenario supuesto. Por ejemplo, un servicio usado dentro de un departamento probablemente tenga un conjunto diferente de políticas del mismo servicio usado por los socios. A este nivel, la definición de clasificaciones internas es importante porque puede ayudar a definir plantillas iniciales conforme a algunos de los escenarios más comunes para promover la coherencia y eficacia. El éxito en esta área no se debe a las selecciones sino a cómo se establece, usa y distribuye la política que los impacta.

Administración de implementación. La mejor manera de promover y cumplir un modelo de régimen consistente para la organización es definir un proceso común de implementación para servicios que brinde valor real. Por ejemplo, si el proceso de implementación automáticamente brinda servicios con soporte

operativo, rápidamente será adoptado. Para fortalecer más su rol, es también importante que este proceso sea flexible y fácil de usar, así los desarrolladores, arquitectos, evaluadores, y operadores de la organización se adherirán al modelo establecido.

Capacidades de la capa soportable.

La perspectiva de implementación del nivel soportable (3) contiene el banco de esquema, política ejecutable, y capacidades para versionar. A continuación las veremos en detalle.

Banco de esquema. Crear y soportar un banco de esquema es el paso siguiente en la activación de un esquema común. Hacerlo permite a los desarrolladores activar y reutilizar definiciones existentes de entidades para *construir* mensajes que se pueden adaptar automáticamente a extensiones de esquema mediante inclusiones e importaciones. Esta activación es un área difícil que pocos clientes pueden manejar con éxito actualmente, pero rápidamente se convertirá en una herramienta invaluable al comenzar a evolucionar los servicios.

Política ejecutable. Mientras el mensaje de solicitud inicia la invocación de un servicio, con frecuencia se desea impulsar la información de política para determinar exactamente lo que se debe ejecutar conforme a ciertas condiciones. Las políticas

"PARA HACER A LOS CONSUMIDORES DE SERVICIOS MÁS AUTOSUFICIENTES, HAY QUE BRINDARLES AMPLIA VISIBILIDAD EN SU USO DE SERVICIOS".

generalmente se piensan como herramienta de uso externo pero las políticas también pueden ser herramientas poderosas para brindar contexto internamente. Si se necesita cambiar el proceso de ejecución o conducta en base a quién realiza el pedido, qué formato necesitan los datos, o la hora del día, la política ejecutable jugará un rol importante en la habilitación para la prestación de servicios.

Versiones. Cómo versionar a los servicios Web es algo que se debe manejar en forma diferente de los típicos enfoques para otro software por su complejidad y extensibilidad. Por ejemplo, para versionar un servicio debe considerarse separadamente del proceso de versionar mensajes del servicio. Afortunadamente, con la flexibilidad inherente de XML y esquemas XML, hay una capacidad de soportar esta complejidad mediante métodos de versión implícitos y explícitos

Aclaración sobre modelos de maduración

Como otros modelos de maduración, ESOMM derivó del modelo de Maduración de Capacidad Carnegie Mellon (CMM), ver Recursos). No obstante, tomamos prestada la noción de un modelo de maduración impulsado por capacidades y nos detenemos ahí. En vez de aplicar orientación al servicio al CMM, tomamos esos principios y los aplicamos a un paradigma de orientación al servicio, esencialmente construyendo un mapa de ruta desde cero. Este modelo no se enfoca en procesos porque el foco está en las capacidades informáticas, no la adopción o preparación organizacional. Al reconocer algunas semejanzas conceptuales con CMM también se ve que ESOMM es una aplicación diferente del concepto de modelo de maduración.

Aunque puede haber valor en algunos de los otros modelos de maduración relacionados con servicios, hay que saber cuánto de la implementación CMM subsume. CMM fue diseñado para tratar los

procesos de una organización relacionados con ingeniería de software, que es un problema muy diferente de definir un plan informático para orientación al servicio. Si se adopta un modelo que conjuga a estos dos, sería abrumador y contraproducente para la organización. Lo que es aún peor, si se adopta un modelo que simplemente renombre algunos de los niveles y capacidades de CMM, se podría estar yendo en la dirección equivocada. Sería como tomar un mapa de Inglaterra con el nombre Londres reemplazado por París y tratar de encontrar la torre Eiffel, simplemente no va a funcionar.

Adoptar este modelo de maduración impulsado por capacidad es una decisión. Elegir el mapa de ruta apropiado es otra. Hay que asegurarse de elegir un mapa que se aplique a los objetivos deseados. Ningún modelo se aplica a todos en todos los casos, hay que asegurarse de elegir uno que lleve a destino correcto.

Figura 3. Capacidades de capas de maduración

	Implementación	Consumo	Administración
Extensible	Colaboración de servicio Orquestación de servicio	SDKs de servicio Política externa	Análisis comercial Administración automatizada de política
Soportable	Versiones Política ejecutable Banco de esquema	SLAs explícitos Portal de servicios Visibilidad de ejecución	Auditoría Monitoreo Modelo de suministro
Repetible	Esquema común Bloques de servicio	Auto suministro Descubrimiento de servicio	Administración de despliegue Políticas empresariales
Utilizable	Patrones de diseño Procesos de desarrollo	Evaluación Contratos explícitos	Modelo de seguridad Monitoreo básico

No obstante, esta capacidad es una característica que las herramientas típicas de desarrollo de servicio no aprovechan y exige algún planeamiento y soporte para su consistente ejecución.

La perspectiva de consumo de la capa soportable es dónde encontrar visibilidad de ejecución, portal de servicios, y capacidades de SLAs explícitas:

Visibilidad de ejecución. Tal vez el único, gran factor en la capacidad de escalar el uso de servicios Web es la capacidad de soportar mayor consumo de servicios brindando capacidades de autoservicio a consumidores existentes y potenciales. Si cada implementación de servicio requiere una serie de llamadas telefónicas o correos electrónicos, algo de matemática mostrará que la capacidad del equipo para desarrollar servicios rápidamente se consumirá por las necesidades de soporte. Un sólo portal que provea acceso a la información correcta puede ser un importante factor para reducir la demanda de los equipos de soporte y desarrollo.

Portal de servicio. Esta capacidad se dirige a un recurso específico al que puede accederse mediante un portal de autoservicio. Para hacer a los consumidores de servicio más autosuficientes, hay que brindarles amplia visibilidad en el uso de los servicios. Más visibilidad se basa en la capacidad de evaluar servicios brindando un nivel de acceso particionado que ayuda a entender qué ocurre durante la ejecución para usar solución de problemas y también deducir si algo no cumple con las expectativas. Esta capacidad compartiría también actividades históricas para ayudar a los consumidores a rastrear la frecuencia con la cual las aplicaciones consumen el servicio.

SLAs explícitos. Para fijar efectivamente las expectativas para los servicios a los consumidores, se debe brindar acuerdos explícitos a nivel de servicio (SLAs). Brindar SLAs es una capacidad difícil porque no existe norma disponible para comunicar esta información para servicios Web. No obstante, se pueden usar diversas formas de documentación o comunicación para compartir en forma eficaz información importante en esta área incluso la expectativa de vida, tiempo de procesamiento anticipado para los pedidos, y cualquier tiempo planificado. Es importante notar que sin un nivel suficiente de visibilidad de servicio y administración a nivel interno, sería casi imposible soportar SLAs.

La perspectiva de administración del nivel soportable contiene el modelo de suministro, monitoreo y auditoría.

Modelo de suministro. El siguiente paso para hacer a los

servicios autosuficientes y soportables es un proceso agilizado para brindar a los nuevos usuarios. Este modelo se usaría para soportar usuarios internos y externos e incluye manejo de pedidos de acceso, creación de nuevas credenciales, y provisión de alertas que notifiquen a los consumidores sobre cambios de servicio o estado de cuenta. Esta capacidad se apoya en la capacidad de auto suministro definida en la capa repetible (2) para brindar un proceso completo de suministro para consumidores de servicio.

Monitoreo. El monitoreo y la visibilidad en la ejecución de servicios es uno de los aspectos más difíciles de soporte de servicios en una escala empresarial debido a la naturaleza distribuida de los servicios Web. La única manera de tratar este soporte en forma efectiva es abarcar los servicios como un sistema, porque el soporte sobre una base "servicio por servicio" constituye un esfuerzo frustrante e ineficaz. Aunque existen muchas maneras de establecer este sistema, se cree que una arquitectura de ejecución común es el medio más efectivo.

Auditoría. La clave para confiar verdaderamente en los servicios que se brindan en la organización es un modelo confiable de auditoría que pueda servir como mecanismo de rastreo para saber qué personas realizaron qué actos y en qué momento. Este mecanismo no sólo ayuda a resolver cualquier controversia que surja, también juega un rol estratégico en un programa de cumplimiento de una empresa.

Capacidades de la capa extensible

Ahora veremos las capacidades de perspectiva de implementación para la capa extensible (4), orquestación del servicio y colaboración del servicio.

Orquestación del servicio. Para activar servicios Web en procesos comerciales bien definidos, se debe tener capacidad para orquestar. Esta capacidad implica flujos de trabajo que pueden

"AUNQUE ES IMPORTANTE SABER DONDE UNO SE ENCUENTRA, LOGRAR UNA ORIENTACIÓN EXACTA ES MENOS IMPORTANTE QUE IDENTIFICAR LAS CAPACIDADES QUE SE NECESITAN TRATAR PARA CONTINUAR AVANZANDO EL VALOR DE LA HABILITACIÓN PARA PRESTACIÓN DE SERVICIOS EN LA ORGANIZACIÓN".

implicar procesos manuales y automáticos pero permite definir un uso consistente y repetible de servicios que se pueden modificar con el tiempo.

Colaboración de servicio. Una vez establecidos los servicios como funciones autónomas que se pueden mantener y soportar, se puede comenzar a mejorar para brindar servicios más complejos y sofisticados. Nos referimos a esta mejora como colaboración de servicio. Mediante la colaboración, se pueden definir los patrones de ejecución sincrónicos y asincrónicos que permiten soportar procesos enteros que constan de servicios existentes. Como ejemplo, se puede tomar un servicio que envía correo electrónico y un servicio que devuelve todos los cambios de horario y exponer un nuevo servicio que envíe un correo electrónico a los clientes notificándoles los cambios de horario.

La perspectiva de consumo de la capa extensible incluye la política externa y las capacidades de SDKs del servicio.

Política externa. Para establecer una infraestructura segura para proveer servicios a consumidores externos y socios, se deben establecer políticas externas, que incluyen no sólo políticas de seguridad soportables externamente, sino las políticas que se han definido para la manera de que estos servicios se puedan unir a los servicios de terceros. Tales políticas incluyen marcas, pago, y términos de uso que son importantes para proteger los intereses del negocio.

SDKs de servicio. Construir un modelo de negocio o un canal mediante servicios Web exige llevar el nivel de soporte para consumidores de servicio al siguiente nivel. Este avance al siguiente nivel puede incluir desarrollo y suministro de kits de desarrollo de software pre-empaquetados (SDKs) que permiten a los desarrolladores el suministro y la implementación rápida del servicio. Estos SDKs potencialmente incluyen componentes UI para uso directo en las aplicaciones del desarrollador y se adaptarían para entornos de desarrollo, ya que no hay estándar para herramientas de desarrollo de múltiples plataformas actualmente.

Finalmente, la perspectiva de administración del nivel extensible incluye la administración automatizada de política y las capacidades de análisis comercial.

Administración de política automatizada. Al comenzar a exponer servicios a un amplio público utilizando las políticas para definir el soporte interoperativo se comienza a recibir pedidos de cambios en la política, por ejemplo, cambios a SLAs o requisitos de seguridad. Estos pedidos se deben procesar en un modo automático. De esa manera no crearán gastos significativos en el escalado de interacción con una base de cliente amplia.

Análisis comercial. El pináculo de la visibilidad en todo el sistema de servicios empresariales es lo que conocemos como análisis de servicio. Esta visibilidad combina los datos técnicos de metadatos de servicio, registros de actividad, y eventos, y permite hacer referencia cruzada de esto con la información de política, datos comerciales, y datos de usuario para brindar excelentes vistas en toda dimensión de los servicios. Esta información se puede utilizar para determinar qué servicios ganan dinero o qué tendencias de usuario están presentes entre los consumidores internos y externos de los servicios.

Aplicación del modelo

Ahora que se tuvo la oportunidad de hacer un breve resumen de los componentes de ESOMM, ya se puede empezar a pensar cómo se puede aplicar a una organización. Para decirlo más claramente, aplicar este modelo no se debe considerar como una actividad de una sola vez o un proceso a corto plazo. En cambio, el modelo funciona mejor como plan de trabajo que se puede modificar con el tiempo al crecer el uso de servicios y la experiencia.

Lamentablemente, el inconveniente de usar el término *maduración* con un modelo es que las personas inmediatamente querrán saber en qué capa están sus organizaciones para entender el sentido de su condición o identidad. Al suceder esto, no hay manera adecuada de responder a la pregunta "¿en qué capa está mi organización?" En vez de dar un grado integral en base a una de las capas, tomamos el enfoque de dar a cada capa su propio nivel de maduración, del uno al cinco, en base a incrementos de a medio punto.

Un nivel de maduración uno significa que no se ha tenido la oportunidad de tratar las capacidades de esa capa, y el cinco significa que se han dominado todas las capacidades definidas en esa capa.

Recursos

Carnegie Mellon Software Engineering Institute
Modelo de Maduración de Capacidades para Software (SW-CMM)
www.sei.cmu.edu/cmm/

Red de Desarrolladores de Microsoft
"Normas de Industria de Arquitectura para Orientación al Servicio", Josh Lee
Microsoft Corporation, 2005.
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbda/html/aisforsoa.asp>

Tabla 1. Un rating ESOM de muestra

Capas de servicio	Nivel de maduración
4 Extensible	1
3 Soportable	1
2 Repetible	2
1 Utilizable	15

Un resultado típico se brinda en Tabla 1. Este rating para cada capa se determina por la fortaleza de su correspondiente capacidad, que va del uno al cinco, la fortaleza con un índice de cinco y la debilidad con uno. El promedio de estas cualidades dará un grado total para la capa, redondeado a medio punto. Por ejemplo, en la capa 1, si se tienen tres de las seis capacidades, medidas como fortaleza, dos medidas como adecuadas, y una como debilidad, la maduración total del nivel será 3.5.

Lo que no se debe obviar en esta discusión sobre rating es el hecho de que ESOMM tiene el objetivo de actuar como mapa de ruta, más que como herramienta de evaluación. Aunque es importante saber donde se está, obtener una orientación exacta es menos importante que identificar las capacidades que se necesitan tratar para continuar avanzando el valor de habilitación para la prestación de servicios en la organización. Mientras se desee cuestionar algunos interrogantes difíciles en forma objetiva en todos los grupos relevantes, se debe tener buena comprensión para los desafíos actuales. Si se aplica la estrategia y objetivos de la organización se debe poder identificar qué capacidades se deben tratar en el corto, inmediato y largo plazo.

Es terriblemente difícil tener una conversación concisa y constructiva sobre la empresa habilitada para la prestación de servicios. La meta es proveerles a las personas como arquitectas de software algunas herramientas y alguna información necesaria para tratar las preguntas correctas al ayudar a la organización a lograr valor real mediante esta poderosa tecnología de habilitación.

Con suerte este breve panorama de ESOMM brindará la suficiente información para comenzar a aplicar algunos de los conceptos en los esfuerzos de habilitación para la prestación de servicios. Cada capacidad fácilmente puede llenar un artículo entero, así que el feedback en cualquier área es bienvenido.

Muchas personas han contribuido a ESOMM y los conceptos presentados aquí, pero me gustaría agradecer especialmente a Wolf Gilbert, Scott Beaudreau, Michel Burger, y Byron Howard por sus aportes. ●

Sobre el Autor

William Oellerman es arquitecto de soluciones en Microsoft Consulting Services, que se dedica a dirigir y dar soporte de oportunidades de habilitación para la prestación de servicios estratégicos en el sector de comunicaciones. William llegó a Microsoft en febrero de 2004 y trabajó más de diez años como arquitecto de software, abarcando consultoría y roles corporativos en Avanade, Ericsson, y Micrografx. Ha sido líder de pensamiento en el espacio de servicios Web durante seis años como orador frecuente en las conferencias de la industria, autor de múltiples artículos en revistas comerciales, y autor de *Architecting Web Services* (APress 2001). Cuando no se encuentra poniéndose al día con las últimas tendencias y desarrollos de habilitación para prestación de servicios, William disfruta ayudando a su mujer a ponerse al día con sus dos activas hijas.

Modelación Orientada al Servicio para Sistemas Conectados – Primera Parte

Por Arvindra Sehmi y Beat Schwegler

Síntesis

Como arquitectos, podemos adoptar una nueva forma de pensamiento para *forzar* esencialmente la consideración explícita en nuestros procesos de diseño de artefactos que modelan el servicio, lo que nos ayuda a identificar de forma adecuada los artefactos y en el nivel correcto de abstracción para satisfacer y alinear las necesidades comerciales de nuestras empresas. Aquí ofrecemos un enfoque de tres partes para la modelación de sistemas conectados orientados al servicio de manera tal que promuevan una alineación estrecha entre la solución IT y las necesidades del negocio. Comenzaremos analizando la perspectiva actual del pensamiento orientado al servicio y explicaremos la forma en la que el pensamiento actual y la conceptualización escasa de la orientación al servicio ha dado como resultado varios fracasos y por lo general, niveles pobres de ganancias sobre las inversiones. Luego, analizaremos los beneficios de insertar un modelo de servicios entre el negocio convencional y los modelos de tecnología que son conocidos por la mayoría de los arquitectos y analizaremos la metodología *Motion* de Microsoft y distribución de capacidades para identificar las capacidades del negocio que pueden asignarse a los servicios. La segunda parte de este artículo se publicará en el Journal 8 y se mostrará la manera de implementar estos servicios distribuidos.

La orientación al servicio ocupa un nivel alto de prioridad en el orden del día de varias organizaciones. El objetivo, por lo general, es adoptar la orientación al servicio para permitir una mejor alineación entre los requerimientos del negocio y los servicios IT dentro de la organización y en última instancia permitir y soportar negocios más ágiles. Sin embargo, en la actualidad, varias organizaciones no logran alinear de forma estrecha los servicios IT con los requerimientos del negocio, y como resultado, no pueden ver los beneficios de las inversiones que se espera que brinde la orientación al servicio.

Al tratar la orientación al servicio, las organizaciones intentan responder las siguientes preguntas: ¿Cómo podemos evitar cometer con arquitecturas orientadas al servicio (SOAs) los mismos errores en los que han resultado iniciativas prometedoras anteriores?, ¿Cómo podemos asegurarnos de que la arquitectura de implementación elegida se relacione con el estado actual o deseado del negocio? ¿Cómo podemos asegurar una solución sostenible que reaccione a la naturaleza de cambio dinámico del negocio?. En otras palabras, ¿Cómo podemos posibilitar y mantener un negocio ágil? ¿Cómo podemos migrar a este nuevo modelo de forma elegante y a un paso que podamos controlar? ¿Cómo podemos realizar este cambio con buen entendimiento en las situaciones en las que se puede agregar el mejor valor para el negocio desde el comienzo?.

Una premisa clave que fomentamos aquí es que para construir sistemas exitosos orientados al servicio es necesario modificar la forma de pensar sobre la orientación al servicio, y el última instancia sobre SOA y sobre la forma de construir sus sistemas. Si no se trata el problema de forma adecuada y se tiene un pensamiento erróneo desde el comienzo podría decirse que es la causa de varios problemas. Sin el pensamiento adecuado, es poco probable que la orientación al servicio cómo técnica proporcione los beneficios esperados.

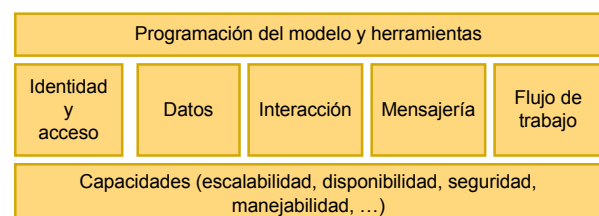
Una parte esencial del pensamiento se refiere a poder conceptualizar el negocio de forma correcta, y al realizar esto, se pueden lograr servicios bien alineados y eficaces que se distribuyen bien sobre las capacidades requeridas por los procesos comerciales. Una vez que se han identificado los servicios adecuados, se puede continuar con la implementación de estos servicios utilizando una selección de tecnologías como las que proporciona la plataforma de aplicación de Microsoft. Analizaremos ahora un modelo que puede aplicarse a los sistemas conectados orientados al servicio para fomentar una alineación estrecha entre las soluciones IT y las necesidades del negocio.

Cinco pilares de los sistemas conectados

La mensajería es importante para la orientación al servicio, pero no es el único aspecto requerido para modelar servicios, también existe una gran cantidad de otros pilares. Si bien la mensajería permite conectar sistemas distintos y proporciona la estructura de conexión subyacente, existe cantidad de otros problemas importantes que deben tratarse y se necesitan pilares adicionales para complementar la mensajería. Analicemos detenidamente los cinco pilares principales de los sistemas conectados (Ver Figura 1):

- **Identidad y acceso.** Este pilar se refiere a la noción de identidad federada (inicio de sesión única en Internet) y la autorización basada en una política. Trata la gestión de relaciones verdaderas y se debe controlar la forma en la que se accede a los sistemas que ahora están conectados. El gobierno y las normas de conformidad representan otros factores importantes que deben ser considerados de forma adecuada bajo este pilar.

Figura 1 Los cinco pilares de los sistemas conectados



Modelación orientada al servicio

- **Datos.** Este pilar trata el agregado de entidades y se refiere a la proporción de una vista armónica y única de una entidad de negocio en particular, como por ejemplo un cliente, a pesar de que los datos del cliente pueden estar duplicados varias veces entre los sistemas múltiples.
- **Interacción.** Este pilar se dedica al consumo de los servicios por parte de las personas –por ejemplo, a través de la creación de interfaces de usuario compuestas y ricas con capacidades online y offline. También se incluyen aquí la interacción en la periferia de las redes a través de Internet, mecanismos punto-a-punto y dispositivos móviles.
- **Mensajería.** Este pilar proporciona una estructura subyacente de los sistemas conectados y requiere soportar la mensajería gestionada, confiable y segura.
- **Flujo de trabajo.** Este pilar trata la automatización o flujo de trabajo de los procesos de negocio, externos a los servicios en sí mismos. El flujo de trabajo se refiere a la organización de los procesos de negocio pero también a otros aspectos como la gestión de interacción del usuario, procesos *ad hoc* y gestión de excepciones.

Si bien el modelo de tres partes que se analiza aquí se refiere en gran parte al pilar de la mensajería, también se consideran aspectos de otros pilares. Antes de analizar un nuevo enfoque para la modelación de sistemas conectados, es importante reconocer que varios de los problemas actuales surgen de un pensamiento convencional que necesita cambiar para llevar a cabo los beneficios verdaderos de la orientación al servicio.

Perspectiva antigua de orientación al servicio

El enfoque tradicional para diseñar y construir soluciones consiste en tomar un conjunto de requerimientos del negocio y a partir de ellos extraer un modelo de tecnología que por lo general requiere orientación al servicio, tecnología de componentes y tal vez sistemas centrales. Si no se logra trabajar estrechamente con el negocio, existe con frecuencia una gran desconexión entre el negocio y las soluciones IT provistas. En vez de encerrarse en sí mismo, es necesario que el desarrollo se convierta en una disciplina de ingeniería con miras hacia el exterior.

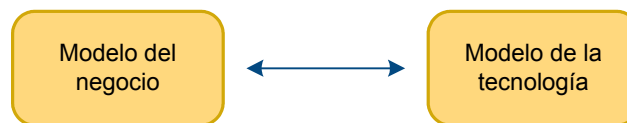
Otro gran problema es que si bien los empresarios pueden ser muy buenos en lo que hacen, no es su trabajo describir lo que hacen a personas que no les interesa lo que hacen, que es realmente lo que sucede cuando hablan con personas IT sobre requerimientos. Desde un punto de vista histórico, los empresarios no han contado con muchas herramientas que los ayuden a crear requisitos objetivos y claros. Por ejemplo, los mapas de procesos son vistas objetivas que no exponen mucha métrica y contenido objetivo –y IT necesita vistas objetivas.

Otro problema con frecuencia ocurre por la forma en la que están organizados los departamentos IT. A veces están organizados de acuerdo a tecnologías específicas –por ejemplo, sistemas centrales, aplicaciones de Internet, aplicaciones de intranet y demás– y otras veces están organizados en torno a firmes barreras de organización humana y límites departamentales. Con ambos enfoques, la organización rápidamente produce *silos* y la manera en la que la información se transfiere es en su mayor parte a través de intercambios de documentos prescritos.

Esta estructura organizacional por debajo de lo óptimo también dificulta la creación de sistemas que incluyen tecnologías múltiples y es muy difícil determinar cuán costosa es realmente una solución debido a la participación de muchas personas de varios departamentos diferentes.

Para colaborar en la solución de esta problema, es necesario que los departamentos IT desarrollen una visión más abierta y más estrechamente conectada con el negocio, y se requiere una nueva forma de

Figura 2 Modelo de un sistema clásico



pensamiento sobre la orientación al servicio y la modelación de sistemas conectados. En el mejor de los casos, el departamento IT debe proporcionar sólo los aspectos de infraestructura de la solución, y el desarrollo de la solución debe estar mucho más firmemente alineado con los grupos de negocios relevantes. La orientación al servicio y los procesos asociados de desarrollo de software no posibilitarán por sí mismos organizaciones orientadas al servicio. Es necesario que sean parte de una estrategia global que organice la transición hacia la IT orientada al servicio.

Modelación clásica de sistemas

Los enfoques actuales de modelación con frecuencia se centran en los modelos de negocio y los modelos de tecnología (Ver Figura 2). Idealmente, existe una alineación y correspondencia estrecha entre el modelo de negocios y el modelo de tecnología, pero en la práctica, esta relación por lo general no es el problema. La razón principal son los departamentos IT internos que no trabajan con suficiente colaboración y efectividad con los negocios. Mientras que los modelos de dominio de la entidad se utilizan por lo general en

"PARA CONSTRUIR SISTEMAS EXITOSOS ORIENTADOS AL SERVICIO ES NECESARIO MODIFICAR LA FORMA DE PENSAR SOBRE LA ORIENTACIÓN AL SERVICIO".

una capa por encima del modelo de tecnología para colaborar en la solución del problema, este enfoque no tiende a funcionar bien, principalmente porque cumple con dos propósitos opuestos: la implementación interna y la exposición funcional del negocio. Es cuestionable que una alineación verdadera entre los modelos de negocios y la tecnología raramente se logra debido a que el espacio entre las dos perspectivas es simplemente demasiado amplio. Por lo tanto, ¿Qué se puede hacer para estrechar la alineación? Consideremos estos cuatro componentes esenciales:

Profesionales IT con miras a lo externo. Los profesionales IT deben volverse más abiertos a lo externo y deben conectarse más profundamente con el negocio de su empresa. Si bien no es necesario que sean expertos en negocios, tienen que poseer un lenguaje objetivo que les permita comunicarse con la empresa sobre los negocios. Los arquitectos en particular, proporcionan los canales de comunicación entre el negocio y los departamentos IT, y deben asegurarse de que los requerimientos y soluciones del negocio sean lo más independiente posibles. La independencia puede lograrse trabajando muy de cerca con el negocio para asegurar que los contratos que ofrece el departamento IT estén mucho más alineados con los requerimientos del negocio. En nuestra opinión, considerar esta perspectiva es la mejor forma (si no es la única) de lograr la granularidad adecuada para los servicios que finalmente se construyen con un enfoque orientado al servicio.

En nuestra lengua vernácula común de hoy en día hablamos de "exponer la arquitectura del negocio". Si bien las capacidades de un negocio para una industria determinada son muy similares, o incluso idénticas, tanto los empresarios como los profesionales IT puede utilizar un conjunto estándar de preguntas para obtener información relevante sobre la arquitectura del negocio y así reunir los requerimientos. Al utilizar un conjunto estándar de

preguntas, aún quienes no son expertos en el dominio de negocios, pueden dirigir un análisis muy útil sobre los requerimientos del negocio y exponer información importante sobre métricas, rendimiento, madurez, interconexión y gobierno y cumplimiento. Debido a que el empresario responde preguntas del arquitecto, el arquitecto colabora de hecho exponiendo la visión que tal vez el negocio aún no posee.

Comprender el cambio de los negocios. Si bien no sugerimos que todo lo que se refiere a IT se convierta en "orientado a los negocios", puede ser de gran valor que IT comprenda la forma en la que el negocio evolucionó en el pasado. Esta comprensión influenciará la implementación de decisiones sobre la provisión de puntos de extensibilidad, versionado, recursos necesarios y asuntos de gestión de un proyecto como cantidad y tamaño de las repeticiones y administración de versiones, incluyendo las versiones principales, secundarias y delta que se esperan en un período determinado de tiempo.

Infraestructura operativa común. Se necesita una infraestructura común para soportar las aplicaciones del negocio que proporcionan prácticas comerciales inter-empresariales e integrales. Es fundamental para el éxito, construir modelos equilibrados para obtener la forma de operar y gestionar la infraestructura e implementar aplicaciones en ella. Las aplicaciones prósperas permiten que quienes toman decisiones comerciales obtengan la comprensión clave de la información que manejan, lo que finalmente permite que los individuos colaboren y tomen decisiones para orientar el negocio de forma más efectiva.

Estándares tecnológicos de servicios Web. Los estándares de los servicios Web permiten que las aplicaciones se conecten. En última instancia, el valor de conectar los sistemas lleva a prácticas de negocios más efectivas y eficientes.

Nueva perspectiva de orientación al servicio

Como arquitectos, ¿Podemos asegurar que los contratos que el departamento IT ofrece al negocio están alineados más estrechamente con los requisitos del negocio? La introducción de

un modelo de servicios entre el modelo de negocios y el modelo tecnológico es un factor clave que puede ayudar a lograr este objetivo. El modelo de tres partes sobre la orientación al servicio que se exhibe en la Figura 3 demuestra una mejora en el modelo del sistema clásico.

Este modelo de tres partes sobre la orientación al servicio interpone un modelo de servicios entre los modelos de tecnología y de negocios que pertenecen al modelo del sistema clásico. La presentación del modelo de servicios muestra diversas ventajas. El

"EL MODELO DE SERVICIOS ES AQUÉL EN EL QUE SE PUEDE CAPTURAR LA SEMÁNTICA NECESARIA PARA EXPRESAR LOS SERVICIOS QUE LOGRAN SOLUCIONES DE ACOPLAMIENTO MÁS ESCASO Y MÁS ABIERTAS A LO EXTERNO O AL NEGOCIO".

modelo de servicios es aquél en el que se puede capturar la semántica necesaria para expresar los servicios que logran soluciones de acoplamiento más escaso y más abiertas a lo externo o al negocio. El modelo de servicios proporciona una ubicación lógica para definir los contratos que aseguran que el lado del negocio de su organización está alineado con el lado IT desde la perspectiva de los requisitos. Al insertar el modelo de servicios, los arquitectos están *forzados* a considerar de forma explícita los artefactos del modelo de servicios en el proceso de diseño.

El modelo de servicios ayuda a los arquitectos a descubrir artefactos en el nivel de abstracción adecuado para satisfacer y alinear las necesidades del negocio. También permite al analista de negocios participar parcialmente del proceso y lograr una mejor trazabilidad de los requisitos del negocio. Por último, los empresarios ya conocen la palabra *servicio*. Por ejemplo, en el contexto de la tercerización, un acuerdo de nivel de servicio (SLA) es bien conocido. Las funciones internas (que no se tercerizan) se comprenden de forma similar, a pesar de que (por lo general) están asociadas con una expectativa de nivel de servicio menos contractual y formal (SLE). En vez de hablar de los servicios y la orientación a los servicios, es mucho mejor hablar sobre los SLEs ya que los empresarios comprenderán su valor.

Un aspecto interesante del modelo de tres partes es que varios aspectos de los cuatro principios de la orientación al servicio, originalmente desarrollados en el nivel de modelo tecnológico, también se aplican a niveles de servicios y de negocios.

Figura 3 Modelo de tres partes de la orientación al servicio

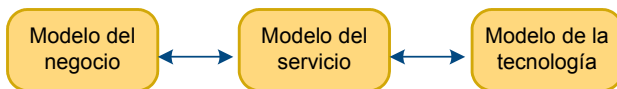


Figura 4 1) Representación de la función del negocio, 2) Modelación de los niveles del servicio y 3) Implementación de los modelos en diferentes niveles de abstracción

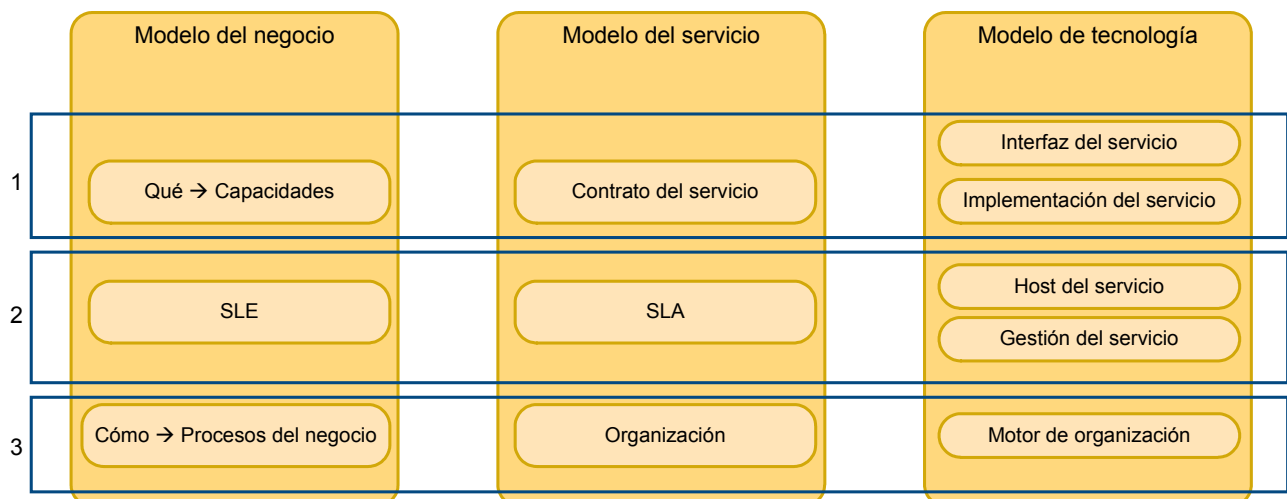


Tabla 1 Los cuatro requisitos del principio y artefactos del modelo

Requisito del principio	Modelo de negocio	Modelo de servicio	Modelo tecnológico
Límite	Capacidades funcionales claramente demarcadas (Análisis gráfico de capacidades en la arquitectura del negocio).	Interfaces de servicios consumibles de forma externa (Definición del punto final del servicio).	Implementación de la periferia del servicio explícita (criterio de valoración concreto del servicio alojado).
Autonomía	Tercerización y utilización de recursos internos para las capacidades (Definiciones de capacidades del entorno, operativas y centrales).	Servicios intercambiables y de acoplamiento escaso (Consideraciones de diseño operativas, técnicas, temporales y espaciales).	Independencia de implementación (Independencia para la implementación de datos y comportamientos).
Contrato	Descripción de procesos y tareas (Unidades lógicas de trabajo y definiciones de flujos de trabajo).	Definición de la organización, interfaz y datos (Mensaje y datos XSD, WSDL y protocolos de interacción de servicios).	Datos que se pueden compartir, interfaz e implementación de la organización (WSDL, XSD y BPML).
Política	Gobierno, SLE y definiciones SLA (Necesidades, preferencias, expectativas y acuerdos).	SLA consumible de forma externa (Definiciones de compatibilidad: declaraciones "Puedo", "Necesito" y "Prefiero"):	Asuntos operativos desacoplados (Tomar atajos afecta la seguridad, confiabilidad, transacciones y WSI-Profile).

El desarrollo orientado al servicio se basa en cuatro principios fundamentales que originalmente eran proporcionados por el equipo *Windows Communication Foundation* (WCF y su código anterior llamado "Indigo") de Microsoft. El equipo originariamente desarrolló los cuatro principios para describir la forma de utilizar el modelo de programación WCF para construir servicios de Web mayoritariamente de grano fino que son más relevantes para el modelo de tecnología que para los otros dos modelos. Sin embargo, nos damos cuenta de que podrían prestar una perspectiva importante al modelo de tres partes cuando se consideran los *principios de servicio empresarial* aplicables a niveles del modelo del negocio y de servicio.

Principios de la orientación al servicio

Los cuatro principios son: los límites son explícitos; los servicios son autónomos; los servicios comparten el contrato y el sistema, no la clase; y la compatibilidad del servicio se determina en base a la política. Analicemos cada uno con más detalle.

Límites explícitos. La orientación al servicio se basa en un modelo de transmisión de mensajes explícitos y cruzar un límite se considera un acto explícito. Para utilizar la analogía de cruzar los límites hacia otro país, el acto de pasar de un país a otro es un acto explícito. Desde un punto de vista del negocio, tiene la misma importancia que comprender y definir los límites funcionales y

"UN BENEFICIO CLAVE AL PRECISAR CON CLARIDAD LA CAPACIDAD DEL NEGOCIO ES QUE MIENTRAS LAS ESTRUCTURAS ORGANIZACIONALES Y LOS FLUJOS DEL PROCESO VIENEN Y VAN, LAS CAPACIDADES ESENCIALES DE LOS NEGOCIOS TIENDEN A PERMANECER CONSTANTES CON EL PASO DEL TIEMPO".

organizacionales dentro de la propia organización para identificar claramente las capacidades demarcadas. Desde la perspectiva del modelo de servicio, este principio requiere interfaces de servicio que puedan ser consumidas de forma externa.

Los servicios son autónomos. Los cambios que se realizan en un servicio no deben impactar de ningún modo sobre otro servicio. Un servicio debe poder volver a escribirse sin impactar de forma negativa a los consumidores del servicio. Continuando la analogía de los países extranjeros, un país puede introducir un nuevo sistema de impuestos independientemente de otro país porque son autónomos. De forma similar, a un nivel de negocios, los cambios

que se realizan de forma interna dentro de un proceso de negocios particular no debe impactar las partes externas que dependen de este servicio o proceso particular. La autonomía a nivel del negocio también significa que los servicios individuales pueden alojarse dentro de una organización o pueden tercerizarse sin un impacto global sobre el proceso de negocios. La autonomía en el nivel del modelo de servicio requiere servicios intercambiables y de acoplamiento escaso. La autonomía a nivel del modelo tecnológico requiere independencia para la implementación.

Los servicios comparten el contrato y el sistema, no la clase. Este principio tiene mucho que ver con que los servicios no expongan sus componentes internos. Por ejemplo, los países publican las normas para completar los formularios que otorgan una visa pero no proporcionan los recursos internos para completarlos. Como otro ejemplo consideremos la interoperabilidad .NET y J2EE. Esta interoperabilidad no es posible si se tratan de intercambiar tipos específicos de plataformas, sino que más bien se requiere un sistema o metamodelo intermedio.

La compatibilidad del servicio se determina en base a la política. En el nivel del modelo del negocio, este principio afecta esencialmente el gobierno y los SLAs y sus definiciones. La política definida a nivel meta describe la capacidad semántica de un servicio en base a un conjunto de definiciones de sus capacidades.

Cada uno de los principios cumple una función importante en el modelo de tres partes e influencia el modelo tecnológico, de servicio y del negocio de diferentes formas. La Tabla 1 sintetiza las necesidades y artefactos para cada modelo en relación con los cuatro principios. Veamos qué define cada modelo y cómo se debe lograr su creación.

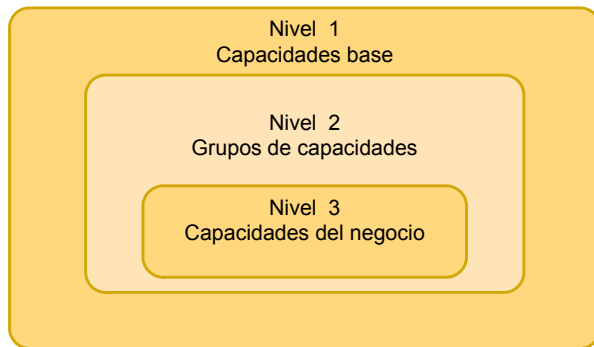
Definición de los modelos

Los cuatro principios son: los límites son explícitos; los servicios son autónomos; los servicios comparten el contrato y el sistema, no la clase; y la compatibilidad del servicio se determina en base a la política. Analicemos cada uno con más detalle.

El modelo de tres partes de la orientación al servicio debe poder representar *qué* es capaz de realizar un negocio en particular –sus capacidades funcionales claves– más que *cómo* actúan las capacidades. Analizaremos brevemente qué es una capacidad de negocio. La forma en la que funcionan los negocios dentro de una organización se representa en diferentes modelos de abstracción a través de tres modelos y se muestra en la Figura 4.

Función del negocio. Desde la perspectiva del modelo del negocio, lo que un negocio es capaz de realizar se expresa identificando las capacidades empresariales. Identificar las

Figura 5: Técnica base para la construcción de un modelo de capacidades



capacidades de una empresa es una tarea fundamental mientras se desarrolla el modelo del negocio. Las capacidades y técnicas para determinar las capacidades de una organización se analizarán más adelante.

Las capacidades funcionales poseen una relación estrecha con la descripción del contrato de servicios definido dentro del modelo de servicios. Más allá del nivel de abstracción, esta relación se traduce a una descripción de interfaces de servicio y en última instancia implementaciones del servicio dentro del modelo de tecnología. (Nótese que si bien existe una fuerte correlación entre una capacidad y un servicio, no es necesariamente uno a uno).

Niveles de servicio. Para asegurar la compatibilidad y la coherencia entre los modelos se presenta la noción de niveles de servicio. En el nivel del modelo de negocios, los SLEs definen los niveles de servicio esperados. Éstos representan las expectativas del negocio. Cuando se construye posteriormente un sistema de software para brindar el servicio, el SLE se traduce a SLA. Al nivel del modelo de tecnología, SLA impacta la forma en la que se aloja el servicio y la manera que se administra el servicio.

Por ejemplo, si las expectativas del negocio o los SLAs son muy exigentes, entonces sería inapropiado alojar el servicio en un servidor con procesador único que no ofrece redundancia. Las expectativas y los acuerdos orientan los requerimientos de disponibilidad, confiabilidad, seguridad y manejabilidad de la solución. De forma similar, si el negocio visualiza el sistema como indispensable, este estado impone responsabilidades adicionales sobre las capacidades de gestión que uno incluye para la solución. La forma en la que se representan los niveles de servicios dentro de cada uno de los tres modelos se muestra en la Figura 4.

En la perspectiva del modelo del negocio, se puede proporcionar a la expectativa tanto información de calidad como de cantidad. La fidelidad desde la información de calidad hacia la de cantidad se vuelve más específica en la medida que uno avanza hacia el modelo de tecnología. Uno pierde información de forma efectiva como parte de las descripciones. Sin embargo, el beneficio de contar con un conjunto de modelos conectados, en particular si pueden describirse en el nivel meta, significa que aún se tiene la capacidad de comprender dónde se originan los requerimientos y qué se relaciona con qué.

Implementación. Es necesario poder describir la forma en la que se implementarán los modelos y poder expresarlos en detalle. Dentro del modelo del negocio, es necesario expresar los procesos de negocio externalizados que se han identificado, y luego es necesario un mecanismo de organización para describir los procesos de negocio en el modelo del servicio. Luego, cuando se implementan las organizaciones, es necesario tener alguna forma de implementación del flujo de trabajo de la tecnología o motor de

organización. Por ejemplo, en la plataforma de Microsoft se pueden utilizar los servicios de organización que brinda el servidor Biz-Talk de Microsoft (Ver Figura 4). Si bien sugerimos flujos de trabajo automatizados cuando hablamos de *organización*, no excluimos los flujos de trabajo humanos o manuales como una forma de *implementación* en el modelo de tecnología, que se vuelve obvia durante la definición del proceso en el modelo de negocios.

Creación de modelos de negocios

Para crear un modelo de negocios eficaz, es necesario poder conceptualizar el negocio e identificar sus funciones comerciales principales. Realizar esto, nos permite lograr servicios bien alineados que se organizan de forma estrecha con los requerimientos de nuestros negocios.

Distribución de la capacidad. Esta técnica fue desarrollada y perfilada por Microsoft para ayudar a conceptualizar la forma en la que funciona un negocio y colaborar en la creación de modelos. Una capacidad de negocio modela lo que realiza una función de negocio individual. No se relaciona con la forma en la que se logra el negocio, sino más bien con su comportamiento visible externamente y sus niveles esperados de rendimiento (es decir, sus resultados). Un beneficio clave al precisar con claridad la capacidad del negocio es que mientras las estructuras organizacionales y los flujos del proceso vienen y van, las capacidades esenciales de los negocios tienden a permanecer constantes con el paso del tiempo. Una capacidad de negocio abstrae y encapsula en un bloque de creación simple a las personas, procesos, tecnologías y procedimientos asociados con una función de negocios determinada. La descomposición del negocio en capacidades proporciona el mayor nivel de desacoplamiento para los contratos de servicio subyacentes.

Pagar a proveedores, despachar productos y generar facturas son ejemplos de capacidades del negocio. En un alto nivel de abstracción, una capacidad del negocio es esencialmente una caja negra con ciertas semejanzas para servicios. Por ejemplo, ambas tienen conexiones que son importantes y están relacionadas pero separadas de sus funciones centrales.

Desarrollar modelos de capacidad. Un modelo de capacidad constituye una jerarquía subdividida de capacidades de negocios que permite modelar el negocio como una red estructurada de capacidades, a diferencia de una red integrada físicamente. Un

"MOTION DESCRIBE LA FORMA EN LA QUE SE EXTRAE Y DOCUMENTA LA INFORMACIÓN ESTRUCTURAL Y ORIENTADA AL PROCESO JUNTO CON LAS CAPACIDADES INDIVIDUALES".

modelo de capacidad de negocio es un diagrama que describe la red de capacidades utilizadas por el negocio. El esquema que se utiliza para construir un modelo de capacidades en diversos niveles de abstracción se muestra en la Figura 5.

El nivel de abstracción con el cual se puede visualizar una capacidad de negocio varía. Las capacidades nivel 1 son las capacidades base comunes a la mayoría de las organizaciones, sin importar el sector de la industria. El nivel 2 y el 3 (y superiores) proporcionan niveles de detalle adicionales sobre las capacidades del negocio. Antes de examinar más detalladamente cada nivel, nótese que no es necesario descomponer todas las capacidades al mismo nivel de refinación; sino más bien, es necesario poner la atención sobre las más importantes para el problema o área del negocio que está en construcción.

Nivel 1 Capacidades Base. Habiendo estudiado el negocio entre varias industrias, Microsoft ha descubierto que a un alto nivel de abstracción los negocios de la mayoría de las industrias exhiben cinco capacidades básicas junto con un grupo de capacidades del

entorno y operativas denominadas en conjunto *capacidades base* (Ver Figura 6).

Las cinco capacidades base comunes a la mayoría de los negocios son el desarrollo de productos y servicios, la generación de demandas, la entrega de productos y servicios, el planeamiento y gestión del negocio y la colaboración. La quinta capacidad, la *colaboración*, es el proceso que organiza y coordina todas las otras capacidades. Se denomina colaboración ya que el proceso puede

“LAS CINCO CAPACIDADES BASE COMUNES A LA MAYORÍA DE LOS NEGOCIOS SON EL DESARROLLO DE PRODUCTOS Y SERVICIOS, LA GENERACIÓN DE DEMANDAS, LA ENTREGA DE PRODUCTOS Y SERVICIOS, EL PLANEAMIENTO Y GESTIÓN DEL NEGOCIO Y LA COLABORACIÓN”.

ser automatizado o manual. El proceso es esencialmente una organización de las capacidades.

Las capacidades base también incluyen *capacidades operativas*, que son las cosas dentro de los límites físicos del negocio de la organización, y *capacidades del entorno*, que representan a todas las otras personas y compañías que interactúan con el negocio y están fuera de los límites físicos del negocio. Estas entidades pueden incluir clientes, socios, proveedores de servicios y autoridades normativas. Son importantes porque todos ellos poseen capacidades que influyen la forma en la que se lleva a cabo un negocio.

Es importante destacar que el límite del negocio no es el mismo que el límite físico de la entidad corporativa. Por ejemplo, si algo se terceriza, es aún parte de la arquitectura del negocio; aunque el trabajo lo realice alguien fuera de la compañía. Existen algunos ejemplos interesantes de trabajos que se trasladan al otro lado de los límites del negocio como el check-in en el aeropuerto, una función que en la actualidad, con frecuencia la realiza el cliente; es la misma capacidad que posee el mismo SLE. Sin embargo, hoy en día existe una tecnología diferente que ha permitido que distintas personas realicen el trabajo, en este caso, el cliente. Esta visión global del ecosistema completo de capacidades permite que las organizaciones vean una variedad más amplia de opciones

que pueden informar los cambios específicos que realizan sobre sus negocios.

Nivel 2 Grupos de Capacidades. Los grupos de capacidades proporcionan el siguiente nivel de detalle en el modelo de capacidades. El nivel 2 es donde comienza el análisis inicial ya que es el nivel que introduce el SLE y los impedimentos y restricciones entre una capacidad y otra; propiedad organizativa; y responsabilidad de gestión. También se identifican las entradas, salidas, funciones de soporte y funciones de control.

Este ejemplo comienza con la capacidad central 3, entrega de productos y servicios, y muestra una jerarquía anidada de capacidades en el nivel 3...n:

3. Entrega de productos y servicios
 - 3.1 Proveer servicio
 - 3.2 Planeamiento avanzado
 - 3.3 Adquisición
 - 3.3.1 Gestión del contrato con el proveedor y contratación
 - 3.3.2 Compra
 - 3.3.2.1 Solicitud de recursos
 - 3.3.2.2 Adquirir/Comprar recursos
 - 3.3.2.3 Gestión de proveedores
 - 3.3.3 Recepción de bienes indirectos/de capital
 - 3.4 Fabricar producto
 - 3.5 Logística

Nótese cómo las capacidades están siempre etiquetadas dentro de su grupo de subordinación adecuado. Un grupo de capacidades es por lo general un nivel inicial importante para realizar un análisis porque es en el nivel del grupo de capacidades que se pueden abstraer y hacer procesables los SLEs, los impedimentos y restricciones, la propiedad organizativa y la responsabilidad de gestión.

Nivel 3 Capacidades de negocios. Los grupos de capacidades se descomponen posteriormente en *capacidades de negocios*. Al organizar capacidades de negocios individuales en un modelo de capacidad (Ver Figura 6), uno termina desarrollando una jerarquía anidada de capacidades de negocios. Las capacidades que están en el nivel 3 y por debajo son los bloques de construcción del modelo.

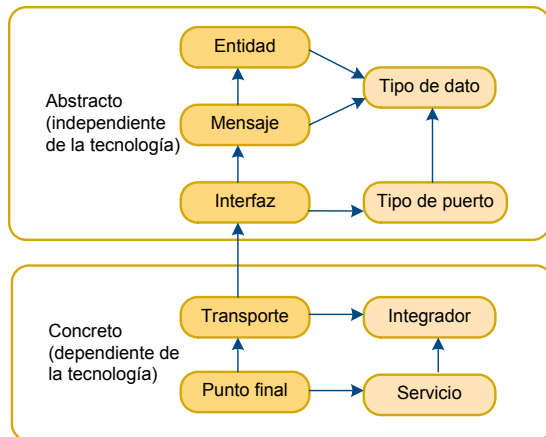
Las capacidades de negocios pueden descomponerse en capacidades de negocios más granuladas –por ejemplo, cuando es necesario definir atributos más detallados. Dentro del análisis se pueden descomponer algunas capacidades de negocio a niveles muy detallados (Nivel 4 y superiores) y se pueden agregar otras capacidades en el mismo nivel. De igual forma, no es necesario descomponer todas las capacidades en el mismo nivel. Se pueden seleccionar y elegir áreas del negocio para modelar en base a los objetivos del negocio en curso y los requerimientos del proyecto.

Para cada capacidad que se identifica en última instancia, se puede definir un conjunto de atributos para describir y documentar la capacidad. Algunos de los atributos claves que son necesarios capturar para cada capacidad incluyen: ¿Quién lo posee? ¿Quién es su cliente? ¿Cuáles son las participaciones y los resultados? ¿Cuáles son los mecanismos requeridos para el manejo de excepciones y la notificación de excepciones? ¿Cuáles son los requisitos de desempeño (pasado, presente y futuro)? ¿Existen implicaciones de gobierno y/o cumplimiento? ¿Impacta el desempeño de la capacidad de forma directa sobre el desempeño de sus capacidades subordinadas? ¿Sobre su departamento? ¿Sobre toda la organización? ¿Qué impulsa a la capacidad a desempeñarse de la forma que lo hace? ¿Las personas? ¿El proceso? ¿IT? ¿Una combinación de estas causas? y ¿Es la

Figura 6: Capacidades base comunes a la mayoría de los negocios



Figura 7: Artefactos del contrato definidos por el modelo del servicio



capacidad parte del motivo por el cual los clientes, socios o proveedores realizan negocios con la organización?

Capturar atributos como estos ayuda a definir el SLE en el nivel de negocios y luego los SLAs en el nivel de servicios. La descripción variada de las capacidades puede pasarse a los grupos de desarrollo quienes pueden utilizar la información para seleccionar las tecnología de implementación apropiadas, hosts y tecnologías de implementación en base al SLE orientado al negocio.

“LAS CAPACIDADES DE NEGOCIOS PUEDEN DESCOMponERSE EN CAPACIDADES DE NEGOCIOS MÁS GRANULADAS –POR EJEMPLO, CUANDO ES NECESARIO DEFINIR ATRIBUTOS MÁS DETALLADOS”.

Capacidades para servicios. Uno de los beneficios claves del enfoque de modelación de capacidades es que permite identificar los elementos estables del negocio para modelar la arquitectura en base a ellos, y proporciona una capa imprescindible que alinea estrechamente la definición de servicios en la arquitectura de sistemas conectados. Se pueden delinear límites en torno a las capacidades –por ejemplo, en el nivel del grupo– y expresar cada capacidad con contratos. Después, se pueden exponer estos servicios y crear una organización entre estas capacidades utilizando los contratos de servicios.

Si bien la organización de las capacidades es una forma eficaz para analizar el estado actual de la empresa, también se puede construir un mapa de capacidades para un estado futuro deseado de la empresa y se puede definir la forma de transformar el negocio para mejorar su agilidad. Como ejemplo consideremos la tercerización. Al dividir las capacidades en aquellas que son centrales para la empresa y aquellas que no lo son, se puede decidir tercerizar a otra empresa las capacidades que no son centrales a través de un proceso de re-ingeniería. Lo importante desde la perspectiva de la empresa es que la arquitectura soporta los cambios en el proceso del negocio sin cambiar la capacidad. La capacidad del negocio total permanece intacta.

Metodología Motion

Microsoft ha desarrollado un método simple, orientado al proyecto que se denomina *Motion* para exponer la arquitectura del negocio por medio de la construcción de mapas de capacidades. Motion es una metodología para organizar, medir y evaluar las capacidades del negocio. Los cuatro elementos de Motion son: herramientas y medidas, metodología prescriptiva, técnicas de organización de capacidades del negocio y modelo de patente en trámite.

La metodología describe *qué* realiza el negocio, representado por las capacidades del negocio, a diferencia de *cómo* lo realiza, representado por personas, procesos y tecnología. Motion ayuda a

¿Cómo funciona Motion?

La metodología Motion de Microsoft no lo estimula a prestar atención a la estructura organizativa de factoring, herramientas o procesos directamente dentro del modelo de capacidades. Sino más bien proporciona una forma efectiva para ayudarlo a lograr soluciones adecuadas para la implementación. La metodología define los siguientes cuatro pasos en el nivel más alto:

- documentando los objetivos del proyecto, luego se genera un mapa de capacidades base y se relacionan los objetivos del proyecto con el mapa de capacidades base.
- paralelos: Capturar vistas del negocio actual, que por lo general comprende la interacción de los empresarios con las personas IT. Capturar detalles de la arquitectura del negocio, en esta etapa se comienza a construir el mapa de capacidades en el nivel superior.
- (*como-está*). En esta etapa se agregan más detalles para crear una arquitectura de negocios.
- puede haber completado el proceso en este punto o tal vez se ha identificado la necesidad de aplicar técnicas de mejora para el proceso, análisis de tercerización del proceso de negocios, y demás.

La metodología a veces se denomina “metodología de desarrollo por etapas” porque existen distintos criterios que uno debe satisfacer antes de continuar hacia la siguiente etapa. También es importante comprender que la metodología motiva la repetición en cada una de estas etapas. Una vez que se han completado las etapas, se utiliza la información que se ha descubierto para construir el modelo posterior. (Ver Recursos para más información).

exponer la arquitectura del negocio de una empresa y aísla los elementos del negocio que definen el modelo de negocios y orienta el desempeño y la diferenciación. El modelo del negocio y la arquitectura del negocio se pueden separar, y Motion ayuda a que las empresas expongan ambas, permitiendo una mejor gestión de cambios.

Por ejemplo, consideremos dos compañías de la industria minorista. Estas compañías poseen las mismas capacidades, pero una tal vez obtenga las ganancias del volumen de ventas de

"EL BENEFICIO PRINCIPAL AL DOCUMENTAR CAPACIDADES DE FORMA DETALLADA Y SISTEMÁTICA ES QUE PROPORCIONA UNA COMPRENSIÓN SÓLIDA DEL SLE QUE EL NEGOCIO POSEE PARA CADA CAPACIDAD DE NEGOCIOS".

productos y la otra genera todo su dinero de una cuota de membresía que le cobra a sus clientes. Si bien la industria es la misma, ambas compañías deben tener modelos de negocios diferentes. Aplicar a una compañía las mejores prácticas de la industria minorista creará valor en una organización y lo destruirá en otra. Por lo tanto, es muy importante tener un entendimiento detallado y claro del modelo de negocios.

Motion describe la forma en la que se extrae y documenta la información estructural y orientada al proceso junto con las capacidades individuales. Esto proporciona la orientación para realizar la actividad de modelación en sí misma e incluye una gran cantidad de plantillas para ayudarlo a documentar toda la información requerida en torno a cada capacidad, (Ver la nota del recuadro lateral "Cómo funciona Motion").

La metodología también proporciona orientación para mostrar la forma en la que se complementan una cantidad de esquemas de mejoras de procesos de negocio existentes, incluyendo los programas Zachman, Lean, Six Sigma y la re-ingeniería de procesos de negocio. Motion no sólo realiza cosas que este modelo no hace, sino que también colabora con las cosas que los otros modelos nunca tuvieron la intención de realizar.

La clave para comprender el motivo por el cual Motion es diferente reside en entender la distribución de capacidades y la manera en la que difiere de la organización del proceso. La arquitectura de las capacidades, y no la arquitectura del proceso, se encuentra en el centro de la metodología Motion. Al abstraer inicialmente (aún ignorándolos) los procesos y analizar las capacidades del negocio, se puede obtener una vista más estable del negocio de forma intrínseca que es particularmente de valor desde el punto de vista del versionado. Posteriormente, las capacidades se convierten en bloques de creación para los procesos.

Si miramos más allá de las prácticas actuales, muchas de las cuales no soportan el ritmo rápido de cambio inherente dentro de los negocios de hoy en día, la metodología Motion se desarrolla como una metodología de arquitectura para los negocios que puede controlar los desafíos presentes y futuros de la economía conectada.

La metodología Motion define aproximadamente 80 atributos que ayudan a documentar las capacidades. Algunos de los atributos que se deben documentar para cada capacidad se han enumerado anteriormente en el análisis del nivel 3 de capacidades de negocio.

El beneficio principal al documentar capacidades de forma detallada y sistemática es que proporciona una comprensión sólida del SLE que el negocio posee para cada capacidad de negocios. Posteriormente, este entendimiento permite obtener SLAs que a su vez ayudan a determinar la tecnología de implementación y la topología de utilización más adecuadas para la implementación del servicio. Contar con una descripción completa y detallada permite

que los equipos técnicos implementen la solución de forma adecuada.

Creación de modelos de servicio

Dado un modelo de negocios, el problema es ahora ¿Cómo se puede traducir este modelo de negocio en un modelo de servicio que se pueda implementar en última instancia? Antes de analizar un enfoque que permite realizar esta traducción, consideremos lo que se necesita definir para crear un modelo de servicio (Ver Tabla 1). En la Figura 7 se muestran los artefactos de contratos claves que son los resultados del diseño y el análisis orientado al servicio y son capturados por el modelo de servicios. Es importante notar que sólo es necesario capturar los artefactos abstractos (independientes de una tecnología específica) para el modelo del servicio. El punto final y transporte subyacente se definen en el modelo de tecnología.

En un nivel abstracto es necesario comprender e identificar las entidades, mensajes e interfaces. ¿Cuáles son las *entidades* contenidas dentro de los datos que se pasan como mensajes durante la interacción entre las capacidades del negocio? ¿Cuáles son los *mensajes* que necesitan fluir entre los diferentes sistemas? ¿Cuáles son las *interfaces* que exponen las capacidades del negocio y, en última instancia los servicios?

En el nivel concreto, el modelo de tecnología define entonces cuáles son los transportes subyacentes y cómo se deben exponer los puntos finales de una capacidad particular, que en última instancia, se traduce a un servicio. Dentro del modelo del servicio y desde una perspectiva contractual del lenguaje de definición del servicio, también es necesario comprender e identificar los tipos de datos y puertos.

Para identificar y documentar los ítems precedentes en un modelo de servicio, no es necesario utilizar radicalmente nuevas técnicas de análisis. Por el contrario, se pueden utilizar técnicas existentes como por ejemplo técnicas de diseño y análisis orientadas al objeto convencional (OOAD) que se aplican en diferentes niveles de abstracción para dar cabida a la orientación al servicio más que la orientación al objeto.

Con el pragmático diseño y análisis orientado al servicio (SOAD), crear modelos de servicios no requiere completamente nuevas tecnologías y enfoques. Más bien, se pueden reutilizar las técnicas existentes y en particular aquellas asociadas con UML y OOAD convencional. Sin embargo, se necesita un cambio de énfasis. Deben alejarse del pensamiento sobre interacciones de llamadas a métodos del estilo RPC entre los objetos y comenzar a pensar en interacciones entre servicios basadas en mensajes. Al realizar esto, se pueden obtener OOAD basados en UML clásicos y aplicarlos al nivel de abstracción del negocio para crear los modelos de servicios. La diferencia principal entre SOAD y OOAD es que SOAD separa claramente el proceso y la entidad para desacoplar servicios. Los servicios desacoplados son más ágiles y reflejan la realidad de las verdaderas prácticas de negocios.

Por ejemplo, consideremos la forma en la que los modelos UML convencionales pueden aplicarse al diseño y análisis orientado al servicio:

- Los casos de uso que incluyen diagramas de actividades se extraen fácilmente de las descripciones de actividades y tareas proporcionadas por el modelo de capacidad.
- El modelo de colaboración proporciona un buen entendimiento del proceso que es necesario representar entre las tareas y las actividades, lo que ayuda a llegar a los requerimientos de organización.
- El modelo de interacción o diagrama de secuencia proporciona información acerca de los patrones de intercambio de mensajes subyacentes (solicitud/respuesta sincrónica, dúplex asíncrono y demás) que necesitan ocurrir entre los servicios.

También ayuda a comenzar a derivar el modelo de dominio canónico externalizado, que define el esquema para los datos en línea que van hasta el mensaje. Con esta información se puede comenzar a definir los contratos de servicios.

- El modelo de componente que se ha bosquejado en una etapa anterior, ayuda a enfocarse sobre las necesidades y problemas operativos, teniendo en cuenta la estructura organizacional y los propietarios y responsabilidades de la capacidad (servicio). Captura información importante para la disponibilidad, confiabilidad y escalabilidad desde una perspectiva de negocios y ayuda a revisar el SLE y los SLAs.

Al utilizar un enfoque SOAD pragmático, se pueden extraer todas las piezas necesarias que se requieren para construir un modelo de servicios. Esta extracción incluye contratos de servicios, SLAs derivados desde SLE definidos por cada capacidad de negocio y los requerimientos de la organización del servicio. Con un

"MOTION PROPORCIONA LA ORIENTACIÓN PARA REALIZAR LA ACTIVIDAD DE MODELACIÓN EN SÍ MISMA E INCLUYE UNA GRAN CANTIDAD DE PLANTILLAS PARA AYUDARLO A DOCUMENTAR TODA LA INFORMACIÓN REQUERIDA EN TORNO A CADA CAPACIDAD".

modelo de servicio detallado, estrechamente alineado y derivado del modelo de negocios, usted se encuentra ahora en una buena posición para asignar el modelo de servicios a un modelo de tecnología que identifique la manera en la que cada servicio se implementará, alojará y utilizará.

Se publicará la segunda parte de este análisis en la próxima edición de *The Architecture Journal*, y se demostrará cómo al utilizar el enfoque precedente y al crear el modelo de servicios, se puede distribuir al departamento IT esquemas de datos, contratos de servicios y los requerimientos del SLA para informar la definición y en última instancia la implementación del modelo de tecnología.

Una nueva dirección

La antigua forma de pensar respecto de la orientación a los servicios no funciona y se requiere el nuevo pensamiento. Al adoptar este nuevo tipo de pensamiento, como arquitectos podemos *forzar* una consideración explícita de los artefactos para el modelo de servicios en el proceso de negocios, lo que lo ayudará a identificar los artefactos de forma adecuada y el en nivel correcto de abstracción para satisfacer la introducción de un modelo de servicios entre los modelos de tecnología y de negocios con el objeto de fomentar una alineación mucho más estrecha de los servicios con las necesidades del negocio. Con un modelo de

Recursos

"Modelación y Mensajería para Sistemas Conectados," Arvindra Sehmi y Beat Schwegler
Un Web cast de una presentación en una Cumbre de Arquitectura Empresarial – Barcelona
(FTPOline.com, 2005) www.ftponline.com/channels/arch/reports/easbarc/2005/video/

Solicite un caso de estudio sobre la Metodología Motion de Microsoft enviando un correo electrónico a motion@microsoft.com.

servicio detallado, estrechamente alineado y derivado del modelo de negocios, usted se encuentra en una buena posición para asignar el modelo de servicios a un modelo de tecnología que identifique la manera en la que cada servicio se implementará, alojará y utilizará. La asignación de capacidades y la metodología Motion proporcionan una forma efectiva de identificar las capacidades de negocios y en última instancia los servicios. La descomposición del negocio en capacidades proporciona el nivel más alto de desacoplamiento para los contratos de servicios subyacentes, y no al revés, como suele ser común en la actualidad.

Los sistemas conectados son instancias del modelo de tres partes, y ellos respetan los cuatro principios de la orientación al servicio. Pueden implementarse de forma *más completa* al utilizar los cinco pilares de las tecnologías de plataforma de Microsoft. Recuerden que inicialmente preguntamos: ¿Cómo podemos evitar cometer con las arquitecturas orientadas al servicio (SOAs) los mismos errores en los que han resultado iniciativas prometedoras anteriores?, ¿Cómo podemos asegurarnos de que la arquitectura de implementación elegida se relaciona con el estado actual o deseado del negocio? ¿Cómo podemos asegurar una solución sostenible que reaccione a la naturaleza de cambio dinámico del negocio? –en otras palabras, ¿Cómo podemos posibilitar y mantener un negocio ágil? ¿Cómo podemos migrar a este nuevo modelo de forma elegante y a un paso que podamos controlar? y ¿Cómo podemos realizar este cambio con buen entendimiento en las situaciones en las que se puede agregar el mejor valor para el negocio desde el comienzo?.

La orientación al servicio con los servicios Web es sólo la implementación de un modelo particular. En la calidad y el fundamento del modelo que determina las respuestas a estas preguntas.

Agradecimientos

Los autores desean agradecer a Ric Merrifield (Director, Microsoft Business Architecture Consulting Group), David Ing (arquitecto de software independiente), Christian Sèller (arquitecto, Thinkecture), Andreas Erlacher (arquitecto, Microsoft Austria), y Sam Chénaur (arquitecto, Microsoft Corporation) por sus comentarios sobre las versiones preliminares de este artículo. También queremos expresar nuestro reconocimiento a Alex Mackman (tecnólogo principal, CM Group Ltd.), un excelente investigador y escritor que nos ha ayudado enormemente. •

Sobre el autor

Arvindra Sehmi es jefe de arquitectura empresarial en el grupo EMEA Developer and Platform Evangelism de Microsoft. Centra su atención sobre la adopción de mejores prácticas de ingeniería de software empresarial a través de la comunidad de arquitectos y desarrolladores de EMEA para la industria de servicios financieros. Arvindra es editor emérito de *The Architecture Journal*. Posee un doctorado en ingeniería biomédica y un master en negocios.

Beat Schwegler es arquitecto para el grupo EMEA Developer and Platform Evangelism de Microsoft. Brinda soporte y asesora a compañías empresariales sobre arquitectura del software y temas relacionados y es orador frecuente en conferencias y eventos internacionales. Posee más de 13 años de experiencia en arquitectura y desarrollo de software profesional y ha formado parte de una gran variedad de proyectos, que abarcan desde el control de la producción en tiempo real y productos de gran venta en dispositivos compactos hasta sistemas ERP y CRM de alta escala. En los últimos 4 años, el interés principal de Beat ha estado en las áreas de la orientación al servicio y los servicios Web.

